电脑学习机上机指导系列丛书之三

游戏 BASIC 语言

于 春 张新莲 编著

電子工業出版社

内容简介

本书是电脑学习机上机指导系列丛书的第三册。F BASIC 语言是一个增强型的、面向游戏的计算机高级语言，因此通常又称为游戏 BASIC。本书面向 BASIC 语言的初学者，从指令、语句、程序等概念出发，从编写九九表的程序开始，通过大量的示例，循序渐进、步步为营，把读者引向深入，最终编写出解决较复杂问题的程序。

游戏 BASIC 语言

于　春　张新莲　编著

责任编辑　施玉新

特约编辑　李　明

*

电子工业出版社出版

北京市海淀区万寿路 173 信箱(100036)

电子工业出版社发行　各地新华书店经销

北京大中印刷厂印刷

*

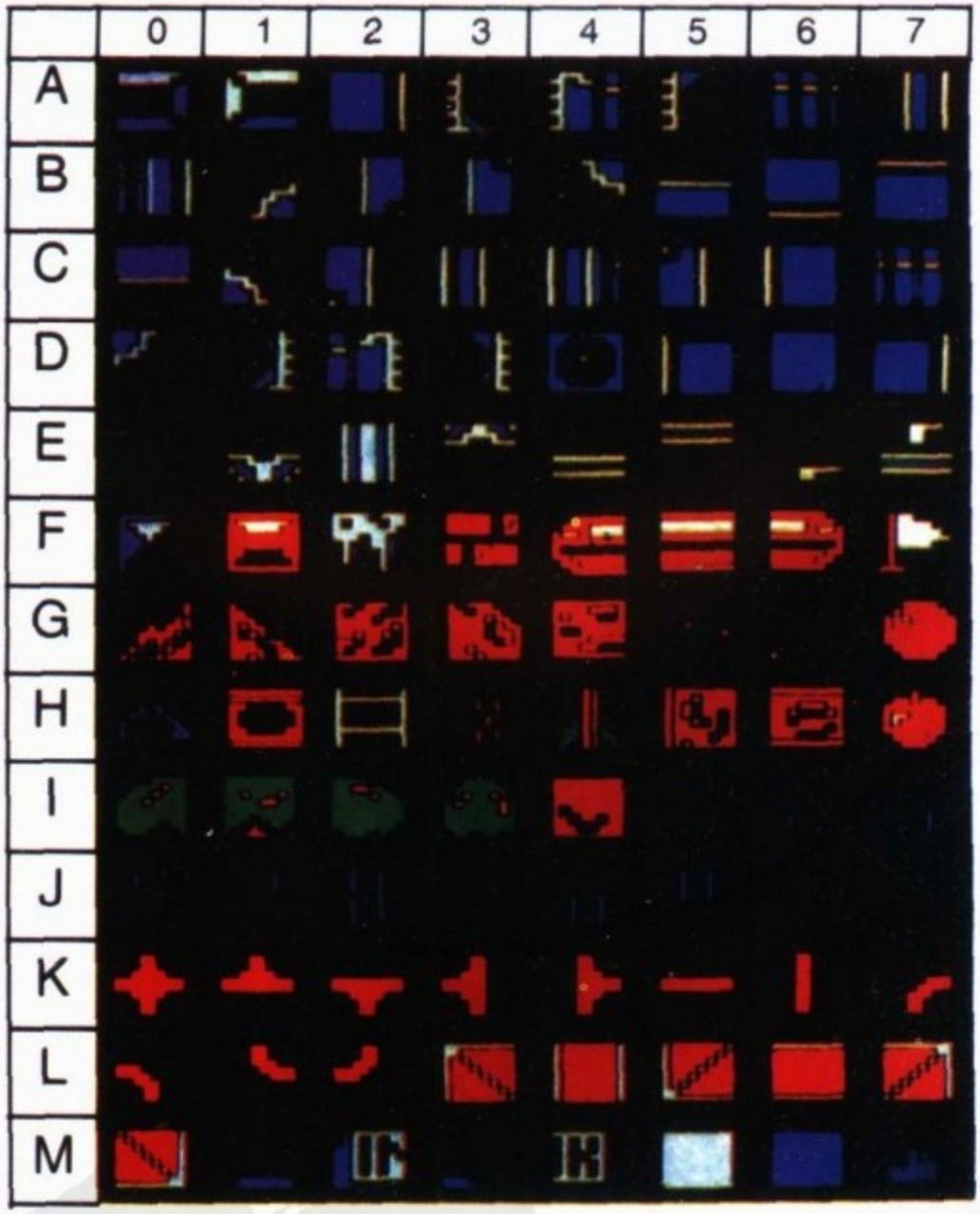
开本：787×1092 毫米 1/32　印张：8.625 字数：198 千字

1996 年 1 月第一版　1996 年 1 月第一次印刷

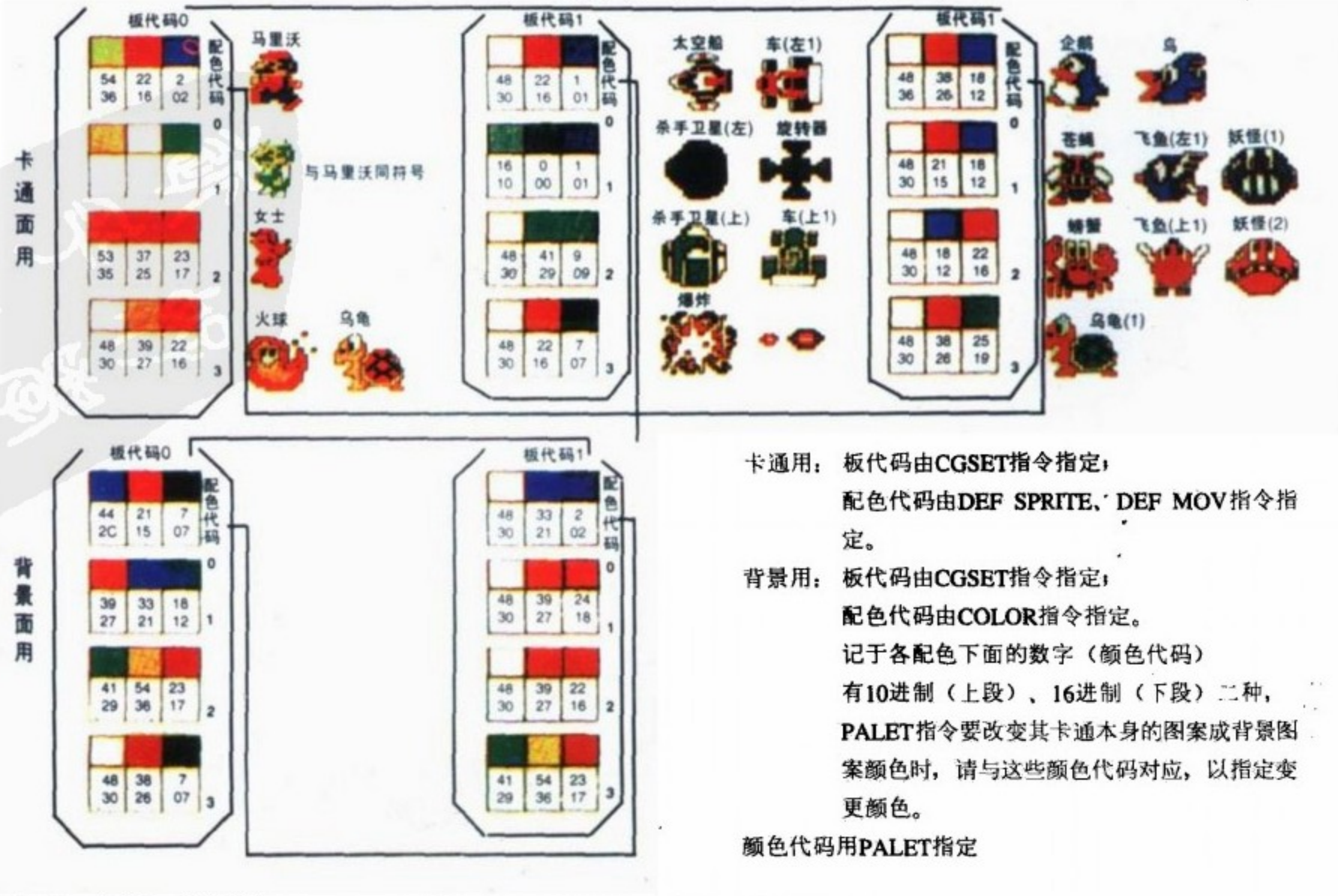
印数：5000 册　　定价：9.50 元

ISBN 7-5053-3070-5/TP·1070

背景图形库



系统配色代码表



卡通用：板代码由CGSET指令指定；
配色代码由DEF SPRITE、DEF MOV指令指定。

背景用：板代码由CGSET指令指定；
配色代码由COLOR指令指定。
记于各配色下面的数字（颜色代码）有10进制（上段）、16进制（下段）二种，PALET指令要改变其卡通本身的图案成背景图案颜色时，请与这些颜色代码对应，以指定变更颜色。

颜色代码用PALET指定

前　言

家庭电视游戏业的发展，突飞猛进、一日千里。随着游戏机的普及，各种以任天堂系列 8 位游戏机为主机、配备键盘而构成的“普及型家庭电脑学习机”也迅速推出。目前生产厂已有三十余家，产品品种近百个。较有名的品牌有：中山小霸王、兰州金字塔、北京裕兴以及深、广一带的金钥匙、和声、科达、科特、美猴王、宏源、智源等，产品档次也已拉开了低、中、高三个级别。有关资料表明，目前我国电脑学习机的拥有量已逾 800 万台。

现在，学习机的功能日趋完善，实用性越来越强。除小霸王还继续维持其寓教于乐的游戏特点以外，各大生产厂家都向提高学习机的实用性进军。其工作系统已由从日本引进时的单一的 F BASIC（整数）语言发展到中文 BASIC（浮点）语言、LOGO 语言，汇编语言、中文编辑、文字处理等系统。就硬件来说，94 年初裕兴率先在其 YX4 型机上推出标准微机 101 键盘，随后和声、科达、金钥匙等立即推出了仿 101 键盘的产品，小霸王、金字塔也于 10 月份推出了仿 101 键盘的 486B 和 9588 产品；金字塔在 94 年率先推出了外接中华机 140K 容量的 5.25 英寸软盘驱动器之后，95 年新春伊始裕兴又推出了与微机完全兼容的 3.5 英寸标准软驱。从应用软件来看，发展更加迅速，已推出了小学、中学乃至高中的各年级辅助教学软件，特别是裕兴推出的仿“金山北大 WPS 6.0”的“裕兴 WPS 文字处理系统”，不仅操作与微机一致，而且文件可与微机相互交流。从而，使学习机的档次又升一级，使它变成了名符其

实的“家庭电脑”(FamilyComputer),完全可以充当现代家庭管理、办公自动化管理的廉价微机。

为了使电脑初学者能够迅速入门,进而登堂入室。我们依据循序渐进的原则,编写了这套丛书。本套丛书分:

操作入门
文字处理
游戏 BASIC **语言**
中文 BASIC **语言**
LOGO **语言**
汇编语言
电脑作曲
电脑绘图
BASIC **语言程序集**
硬件与维修

共十册。

本书是丛书的第三册。

F BASIC 语言是一个增强型的、面向游戏的计算机高级语言,因此通常又称为游戏 BASIC。

同一般 BASIC 语言相比,F BASIC 有功能更强大的指令系统、方式更灵活的程序结构、操作更简单的动画控制、音色更优美的旋律演奏和色彩更丰富的背景设计。这些特点可从任天堂游戏中全面体现出来。

F BASIC 有多种版本,本书选择了目前所有电脑学习机都配备的、最基础最易掌握的 BS. 2A 版本进行介绍。

本书面向 BASIC 语言的初学者,从指令、语句、程序等概念出发,从编写九九表的程序开始,通过大量的示例,循序渐进、步步为营,把读者引向深入,最终编写出解决较复杂问题

的程序，使读者在不知不觉中步入 BASIC 的殿堂。

本书的由苏子栋教授审定。在编写过程中，陈建宇、韩玉文、李观勇、于勇、等同志作了大量的工作，在此一并表示感谢。

编　者

1995 年 1 月于北京

丛书学习说明

本丛书是面向电脑初学者的科学普及读物。丛书的编写结构安排，从初学者最易接受的角度及学习兴趣循序渐进、逐步深入，引导初学者在不觉中步入电脑的殿堂。

丛书的前五册为基础篇，重点介绍了电脑基础知识、电脑学习机的选购常识以及键盘、软盘驱动器、打印机等微机外设的基础知识；常用汉字输入法和中文文字处理操作方法；F BASIC、中文 BASIC、LOGO 三种计算机语言的指令及编程方法。

丛书的后五册为提高篇，重点讨论了 6527CPU 工作系统的特点、汇编语言的指令及编程方法；音乐板、PLAY 指令、机器语言三种电脑作曲方法；键控、BASIC 指令、机器语言三种绘图方法；精选了 60 个 F BASIC、中文 BASIC 语言程序；最后讨论了电脑学习机的硬件结构原理和基本维修方法。

整套丛书内容连贯，但又独立成章。十本书读者可依序学习，也可在学习了丛书之一《操作入门》后，根据自己的兴趣和爱好挑选学习。如需要进行文字处理的用户，只要再学习丛书之二就可胜任；爱好音乐的朋友可继续学习丛书之七，就可令电脑演奏音乐，若要进一步提高，还须学习丛书之六，使用机器语言编程才能演奏美妙的旋律；电脑绘图爱好者，则要在丛书之三、之八上下功夫；许多朋友在学习了游戏 BASIC 语言后，想自己动手编写任天堂游戏程序及各种实用程序，那么丛书之九提供了各种类型的程序范例供你参考。

本丛书的撰写宗旨就是辅导初学者掌握电脑的基础知

识，有深造要求的朋友，请参阅《F BASIC 语言与编程技巧》、《电脑游戏机硬件与编程特技》、《任天堂游戏编程探密》。

目　录

第一章　进入 F BASIC 的大门 …… (1)

第一节　什么是程序 …… (1)
第二节　什么是指令 …… (5)
第三节　输入输出的概念 …… (7)
第四节　语句的概念 …… (10)
第五节　F BASIC 中的变量 …… (12)
第六节　屏幕编辑 …… (14)
第七节　本章小结 …… (17)

第二章　从“九九”歌开始 …… (19)

第一节　电脑编写九九歌诀 …… (19)
第二节　循环语句 …… (20)
第三节　循环语句的应用 …… (22)
第四节　多重循环 …… (24)
第五节　转向语句 …… (27)
第六节　指令与程序 …… (31)
第七节　本章小结 …… (35)

第三章　梁山好汉排座次 …… (46)

第一节　显示按键上的字符 …… (46)
第二节　输入一串字符 …… (47)
第三节　键盘输入成绩排名次 …… (52)

第四节　程序输入成绩排名次 …… (56)
第五节　多科成绩排名次 …… (58)
第六节　排序程序的优化 …… (63)
第七节　本章小结 …… (65)

第四章　飞奔的玛丽翱翔的鹰 …… (74)

第一节　迷人的卡通图案 …… (75)
第二节　独特的显示方式 …… (76)
第三节　潇洒的玛丽兄弟 …… (78)
第四节　让玛丽哥哥扭起来 …… (83)
第五节　玛丽哥哥走八角 …… (91)
第六节　服从命令听指挥 …… (94)
第七节　本章小结 …… (103)

第五章　七彩缤纷的画面…… (115)

第一节　屏幕万花筒 …… (118)
第二节　诱人的金山 …… (118)
第三节　神奇的设计师 …… (131)
第四节　本章小结 …… (145)

第六章　惊人的运算精度…… (151)

第一节　运算方法的引入 …… (151)
第二节　基本子程序 …… (153)
第三节　高精度加法运算 …… (157)
第四节　高精度减法运算 …… (162)
第五节　高精度加减混合运算…… (166)
第六节　高精度乘法运算 …… (172)

第七节　高精度除法运算 …… (176)
第八节　本章小结 …… (182)

第七章　趣味程序集锦…… (187)

第一节　还有什么指令 …… (188)
第二节　格式规范的“九九表”…… (208)
第三节　勾股定理的验证 …… (209)
第四节　算算看 …… (211)
第五节　谁是罪犯 …… (214)
第六节　大赛排名次 …… (215)
第七节　方便的电话号码册 …… (218)
第八节　电脑猜谜 …… (220)
第九节　数字魔方 …… (226)
第十节　考秀才 …… (229)
第十一节　哥德巴赫猜想 …… (234)
第十二节　小企鹅进城 …… (240)
附录一　卡通图形库代码一览表…… (244)
附录二　背景图形库代码一览表…… (248)
附录三　ASCII 码表…… (252)
附录四　背景图形与代码对照表…… (254)
附录五　F BASIC(BS. 2A 版本)出错信息一览表 … (255)
附录六　F BASIC(BS. 2A 版本)指令汇总表 …… (256)
附录七　背景图形库(封二)
附录八　系统配色代码表(封三)
附录九　卡通图形库(封四)

第一章
进入 F BASIC 的大门

未接触过电脑的人总认为：电脑是高科技的产物，结构神秘、操作复杂，似乎远不可及、高不可攀；尤其一听到“指令”、“程序”、“硬件”、“软件”、“计算机语言”等术语就头大，更感到电脑是一般人不能插足的神圣领域。

“不识庐山真面目，只缘不在此山中”。其实，电脑是极易学习、掌握的工具，它与我们的日常生活有着千丝万缕的联系，它的许多操作与我们的社会活动也有着附影随形的相似。本章就从最简单的数学计算开始，引入“指令”、“程序”、“语句”等概念，带领大家步入 F BASIC 语言的大门。

第一节　什么是程序

学习计算机语言首先遇到的名词就是“程序”，那么“程序”是一个什么样的概念呢？

初学电脑的朋友都觉得较难理解。

下面我们通过例题的计算明确“程序”的概念。

例 1—1　李明的期中考试成绩，数学 96 分、语文 88 分，他的总成绩是多少分？

计算：

列横式：　　96＋88＝？

列竖式：

$$\begin{array}{r} 96 \\ +\ 88 \\ \hline 184 \end{array}$$

答案是：　　96＋88＝184

用文字表达就是：96 加 88 等于 184。

这是我们在数学中的计算方法，似乎与电脑挂不上钩。下面再做另一方法的计算。

我们都已熟悉了用电子计算器进行简单的数学运算。对于李明的期中总成绩，若用计算器计算，操作步骤如下：

依次按

9、6、＋、8、8、＝

六次按键后，将在计算器的液晶显示屏上显示 184。

若把上述操作步骤用文字叙述，则是：

按 9、6 键————把数学成绩 96 输入计算器

按“＋”键————通知计算器做“加法”运算

按 8、8 键————输入语文成绩 88

按“＝”键————输出计算结果 184

那么，电脑是如何计算的呢？

现在，请打开电脑电源，按手册说明，进入 F BASIC 系统工作状态。这时，屏幕上应出现

```
FBASICBS. 2A
4030BYTESFREE
OK
■(闪烁的方块)
```

闪烁的方块叫做光标，它表示电脑正处于工作待命状态。

请键入如下内容：(即按相应的字母键，电视屏幕上应显示相应的字母)

```
? 96+88
184
OK
■
```

"↙"表示按"Enter"键，该键称为输入键，在这个例子里，它相当于计算器上的"="号键(俗称"回车键"，它是根据打字机的操作而引入的术语。在打印机上，打字机的打印车回到起始位置就叫"回车"。)

当输入了以上 8 个字符，按回车键后，将看到屏幕上立即显示出 184 的计算结果。"OK"表示电脑完成了操作，等待新的输入。

对以上输入的内容，我们可以这样理解：

问：96+88 等于多少？

只要按回车键，它就立即告诉得数。

它与计算器操作的不同之处就是多了一个"?"，但显然比计算器优越的多。计算器只能看到结果，看不到已输入的算式；而电脑则算式、结果都看得到。

再计算一题：

例 1－2　张华的期中考试成绩，数学 93 分、语文 85 分，他的总成绩是多少分？

键入如下内容：

```
? 93+85↙
178
OK
```

■

屏幕上立即显示‘178”分。

计算第三题

例 1－3　李明期中考试成绩是 184 分，张华的考试成绩是 178 分，李明的总成绩比张华多多少分？

这是一个减法运算题。请键入以下内容：

？ 184－178↙

6

OK

■

屏幕上立即打印“6”分。

现在，大家是否觉得使用电脑同使用计算器一样简单。

说到这里，我们要返回本节的标题“什么是程序”上来。

那么，什么是程序呢？

我们说，以上输入的都叫程序。

由于这里的每一行程序独立完成一项操作，所以可称为一段程序。

这类程序因为输入电脑后，电脑立即执行，显示答案，所以称为立即执行型程序（后面我们还将讨论另一类程序——暂缓执行型程序）。

显然，程序就是指挥计算机完成某项操作的一串字符。

这就是程序的概念。

顺理成章，我们写程序的过程则叫做编写程序，简称编程。

是不是我们已经尝试过编程了？

第二节　什么是指令

顾名思义，“指令”就是指示、命令。

那么，在我们上一节的操作中有没有用到指令呢？回答是肯定的。

“?”、“+”就是关于输出和相加的指令。

大家可能觉得有点奇怪，“?”、“+”怎么能是指令呢？

原来，为了方便记忆、书写、输入起见，电脑中的许多指令都采用了简写、缩写形式。“?”就是“PRINT”指令的简写。

“PRINT”在英文中是打印的意思，在这里可以理解为“输出”或“显示”（由于 F BASIC 不支持打印机，所以运算结果不能打印出来。）

在以上三段程序中，使用“?”和“PRINT”指令是等价的，大家不妨输入电脑验证一下。

通过验证，大家可能已经明白了“?”是输出指令的含义。

现在可能有人要问，“+”是那个指令的缩写呢？

回答是：“+”不是哪条指令的缩写，它本身就是一条指令。

以上我们使用的“PRINT”指令为命令型指令，它等效于向电脑下达命令。该命令的作用就是让电脑把后面的内容显示在屏幕上。

“+”则为算符指令，它的作用是指挥电脑把该指令前后的两个数据进行求和运算。

下面我们再进行几道题的运算。

例 1—4　李明的期中考试成绩，数学 96 分、语文 88 分，他的平均成绩是多少？

这是一个先求和、再求商的混合运算题。其算式为：

(96+88)÷2=?

计算上题的程序为：

```
? (96+88)/2↙
92
OK
■
```

从上一段程序中可以看出，计算机程序的格式与笔算差不多，不同的是把“÷”改成了“/”，这好象是分数中的分数线。上面的“96+88”是分子、下面的“2”是分母。

理所当然，“/”也是算符指令。

例 1－5　李明在期中考试中，数学、语文平均成绩是 92 分，他的总成绩是多少分？

这是一个乘法运算题。算式为：

92×2=?

计算该题的程序为：

```
? 92 * 2↙
184
OK
■
```

电脑中的乘运算符指令为“ * ”，这是与数学算式的不同之处，希望大家特别注意。

以上我们学习了“+”、“－”、“ * ”、“/”四个运算符，下面认识 F BASIC 中的最后一个算术运算符“MOD”。

“MOD”是求余数的运算符，为便于认识，我们先做一道求余数的运算题。

例 1－6　132 除以 28，余数是多少？

列除法竖式运算如下：

$$\begin{array}{r} \quad 4 \\ 28\ \overline{)\ 132} \\ \quad 112 \\ \hline \quad 20 \end{array}$$

即　132÷28 的余数是 20。

例 1－6 的 F BASIC 求余程序为

? 132 MOD 28↙

20

OK

■

可见，电脑求余运算也象加、减、乘、除运算一样，立即显示得数。

请大家记牢求余运算的程序格式。

指令在 BASIC 语言中也叫做保留字。所谓“保留字”就是说这些字（英文单词或英文字母组合）已被电脑使用，用户只能使用它们作为指令编写程序，不能挪做它用。

第三节　输入输出的概念

一、基本概念

一谈到电脑就离不开输入、输出这两个名词。

何谓输入、输出呢？

实际上，在我们以上的操作中，已多次进行了输入、输出

操作。简言之，我们通过键盘敲入字符就是输入操作；而电脑运算完毕后的结果显示在屏幕上就是输出操作。电脑输出操作的指令就是“?”所发出的。

所以，以后的叙述中，一提到输入某某内容，大家就应该想到是用键盘向电脑输入指令或程序；一提到输出就应想到某种结果将显示在屏幕上。

当然，输入操作不仅仅限于敲键盘，以后将会看到通过录音机把存在磁带上的程序调入电脑是输入操作；而把程序存入录音机磁带的过程则叫做输出操作。

二、并列输出与说明输出

1. **并列输出** 以上计算李明、张华的成绩是分别编程的。有的朋友可能要问，能否一次输出李明和张华的平均成绩呢？

当然可以。如果不能，电脑岂不同计算器一样了。

编程如下：

```
? (96+88)/2,(93+85)/2↙
92      89
OK
■
```

细心的读者已经看出这一程序就是由两条求李明、张华平均成绩的程序合并而成的，不同的是把第二个程序的“?”省略，而在两个算式之间加了一个逗号。

看对了！

这就是 PRINT 指令的特点，也是电脑区别于计算器的地方。

其实，一条 PRINT 指令不只限于输出两项内容，它可以

控制输出多项内容，项数的多少与该段程序的最长限制有关。

F BASIC 规定，一段连续输入的程序（或以后将学到的语句行）最长容纳 256 个字符。所以，只要不超过 256 个字符，“?”后可以跟任意项。如下：

```
? 25+13,18*35,32/8,125*30+2
4*25/6-53,(55+33+22-89+47-
28-13+15678-89)/15
```

等等都可以。

现在是否体会到电脑相对于计算器的优越性了？

2. **说明输出** 在以上第一段程序中，电脑输出了两个数据，由于数据量较小，我们知道第一个数据 92 是李明的平均成绩、第二个数据 89 是张华的平均成绩。但当一次输出的数据较多时将不容易区分。

有没有对输出数据进行说明的方法呢？

当然有。

请输入下列程序

```
?"LiMing="(96+88)/2,"ZhangHua="
(93+85)/2↙
LiMing=92      ZhangHua=89
OK
■
```

这一程序说明，PRINT 指令允许带注释，注释的内容可与计算结果一起输出，但注释必须用双引号括起来。

在 F BASIC 语言中，PRINT 指令功能最强、使用得最频繁、最灵活。在后面的章节还要详细地讨论它。

第四节　语句的概念

以上我们已经知道了什么是程序、什么是指令。知道了立即执行型程序的格式。现在我们再来认识程序的第二种形式——暂缓执行型程序。

首先请原样输入如下程序：

```
1 ? 96+88↙
■
```

输入程序回车后，仅仅是光标移入下一行，没有显示任何结果。为什么会是这样呢？

对照计算李明总成绩的程序，发现它仅在程序的开始多了一个数字“1”。

问题就出在这个“1”上。

这就是暂缓执行型程序的格式。

与立即执行型程序的区别是：暂缓执行型程序输入电脑后，电脑并不立即执行该程序，只是把程序存入电脑的内存中，等待用户输入下一条程序。

现在请继续输入程序

```
2 ? 93+85↙
■
```

当然，电脑只是把程序存入，不输出结果。

这时，可能有的朋友要问，暂缓执行型程序有什么作用呢？

解释这个问题之前，请先熟悉两条指令：

LIST

RUN

“LIST”在英文中为“清单”、“列表”的意思。在F BASIC语言中，它的作用是指挥电脑列出用户已输入内存的程序清单，简称列表。

同“PRINT”指令一样，“LIST”指令也有其简写形式。它的简写是：“L.”。

（注意——“L”后的“.”号是不能省略的）

请输入

L.↙

将显示如下内容：

1 PRINT96+88

2 PRINT93+85

OK

■

列出的程序清单就是我们刚才输入的两行程序，不同的是电脑已把简写指令“?”翻译成“PRINT”指令原形。

现在再来讲“RUN”指令。

“RUN”在英文中是“跑”、“运行”的意思。这里取其运行的含义。它的作用就是命令电脑运行内存中的程序。

“RUN”的简写形式是“R.”。

请输入：

R:↙

将显示

184

178

OK

■

这说明电脑一次把内存中的两个程序执行完毕。

上述以“1”、“2”开头的两行程序就叫做语句，习惯上称为语句行。因此，以“1”开头的就称为“1 行”，以“2”开头的则称为“2 行”。

现在，大家可以感觉到暂缓执行型语句比立即执行型语句的优点了(以后将会进一步发现它有更多的优点)。

首先，程序可以存储，不必输入一行程序就必须运行一行；

其次，一次输入的程序可以反复运行(大家可再执行几次“RUN”，将会看到，每执行一次电脑就输出一次结果)；

第三，欲进行的运算结果可以集中显示。

从以上讨论中，我们已经有了暂缓执行型语句的一般概念，就是每一行语句由一个数字作引导，其后跟着指令和表达式。为方便叙述，一般把程序前的数字叫作该语句的标号或行号。

F BASIC 规定，语句行的标号必须是正整数，取值可以是 0～65534 中的任何数值。为方便修改程序、插入语句行，语句行的标号一般取 5 或 10 的倍数为间隔，如 10、20、30 等等。在输入程序中各语句行不一定顺序输入，电脑会自动按标号的大小顺序排列。

第五节　F BASIC 中的变量

说起变量，大家并不陌生。现在的小学生就已接触了 X、Y 等未知数类的变量。大家也都知道任何单个、多个英文字母或英文字母后加数字都可以做变量，如：

A、AB、ABC、A1、B2、C89、abc、aa

等等都可以做变量。

F BASIC 中的变量与数学中变量的概念差不多，但不识别小写字母，所以变量名不能用小写字母。这类变量一般称为整型数值简单变量，简称整型简单变量或变量。如：

A、A1、A2、ABC、SMD .

等。保留字不允许用作变量。

那么，变量在程序中有何用处呢？

在以上所编的程序中，每计算一次都要改变一次 PRINT 指令后的输出内容。当算式较简单时还可以忍受，但当算式较复杂时若仍计算一次输入一次，将会变得令人不能容忍。如：

? (23 * 34+(34−23) * 34/23)−23

尤其当用同一算式计算不同数据时，更显麻烦。但若在程序中引入变量将很好地解决这一问题。以上例为例，可以这样编程。

```
10 A=23
20 B=34
30 ? (A * B+(B−A) * B/A)−A
```

程序中的 10、20 两行是对变量赋值语句，使用中只要改变变量 A、B 的赋值，就可以用 30 行的一个算式进行多次计算，这显然是极其方便的。如计算李明的期终平均成绩的程序可以这样编写：

```
10 A=96
20 B=88
30 ? (A+B)/2
```

而计算张华的将仍是这一程序，仅把 A、B 的赋值改为 93、85 即可。下一节我们将学习如何修改程序。

另外，F BASIC 中还有另一种变量——字符串变量。

顾名思义，字符串变量的值不是数字而是字符串。“字符

串”当然就是一串字符了。

为了与简单数值变量区别，字符串变量用大写英文字母加美元符号“$”表示。（“$”的读音近似为 duole）如：A$、AB$、A1$、C$等等。

字符串变量有什么用途呢？

记得在说明输出中，我们分别用了“Li Ming”、“Zhang Hua”来说明各自的成绩，有了字符串变量后我们可以把他们的姓名对字符串变量赋值，然后在打印语句中直接使用字符串变量。如：

```
10 A=96
20 B=88
30 A$="Li Ming="
40 ? A$;(A+B)/2
```

这种形式同样能打印出“Li Ming=92”。

现在，可能大家觉得这样做太罗嗦，以后的学习中将会逐渐体会到它的优越性。

第六节　屏幕编辑

所谓“屏幕编辑”就是移动光标和修改程序。

当我们通过键盘向电脑输入命令或程序时，输入的内容也同时显示在屏幕上。若发现输入错误或指令格式不对时，就要进行修改。修改程序就要用到屏幕编辑，这就如同我们修改作文一样。

当作文中有错字时，一般已由老师用红笔标出了错字的位置，我们只须用橡皮擦去错误的字，再写上正确的字。

修改程序也与此类似，只不过是用光标标出欲修改的地

方，以告诉电脑哪里要修改。移动光标通常使用“→、←、↑、↓”四个方向键。箭头的指向就是光标的移动方向。

擦去程序中多余字符的橡皮就是“Del”键。每按一次该键，将擦去光标前的一个字符。

一篇誊清的作文，若发现漏写了字，一般是把漏写的字补写在漏写位置的上方，然后画一个插入符号。编程中若漏写了字符，插入方法是先在漏写的位置挤出一个空位，再把漏写的字符添上。以此保证程序整齐划一。挤空位的按键就是“Ins”键。

屏幕编辑中还经常用到：空格键、Home 键和 Shift＋Home（注——Shift＋Home 键表示先按住 Shift 键不放，再按 Home 键）等键。下面具体介绍屏幕编辑方法。

一、擦去多输入的字符

当欲擦去多输入的字符时，可使用方向键把光标移到多余字符的右边，按“Del”键，就可将该字符擦去，同时光标移到擦去字符的位置。

例如，已输入：

10 PRINTTX■

发现多键入了一个字符“T”，则按方向键把光标移到“T”的右边，有

10 PRINTT■X

按“Del”键后，有

10 PRINT■X

按方向键把光标移到“X”后面，又可继续输入以后的程序。

一般把擦去多输入字符的操作称为“删除”。

二、更改已输入的字符

在输入程序的过程中，若发现已输入的字符是错误的，需要更改时，可使用方向键把光标移到要更改的字符上面，直接输入正确的字符即可。

例如，已输入：

30 PRENT 32+28

发现“PRINT”语句中的第三个字符“I”误键为“E”，则按“←”键，使光标盖住“E”，按“I”键，则修改为：

30 PRINT 32+28

三、增添漏输入的字符

如输入以下程序：

10 ?"Li Mng=";(96+88)/2■

时发现“李明”的“明”的拼音漏输了字母“i”，这时，首先把光标移到“Mng”的字母“n”上，按“Ins”键将在“M”与“n”之间挤出一个空格。

10 ?"Li Mng=";(96+88)/2

输入“i”就完成了修改。

一般把增添漏输入字符的操作称为“插入”。

四、删除多输入的一串字符

有了删除一个字符的经验，删除多个字符并非难事。方法就是把光标键移到该串字符的最右边，按住“Del”键不放，即可连续删除。一直删除到最前一个字符时松开按键。

五、清除整个屏幕

有时，我们并不需要当前屏幕显示的内容，想全部删除后再输入程序，以使屏幕显示清晰。这时，若使用“Del”键将显得太慢。系统提供了一个全屏幕清除的复合键“Shift＋Home”，只要按复合键一次就可把屏幕清除的一干二净。

随便按键，输入一些乱七八糟的东西后，不妨按“Shift＋Home”键试试。

另外，使用“Ctrl＋E”键将删除光标所在的一整行；使用“Ctrl＋Z”将删除光标以后的所有字符。

请自己试一试。

六、光标的快速移动

分别按住“→、←、↑、↓”键可实现光标向四个方向快速移动。按“Home”键时，则不论目前光标在何处，都可直接移到屏幕的左上角。

总之，屏幕编辑方法灵活多样，大家先学会以上最基本的编辑方法。随着学习的深入和练习时间的积累，必然“熟能生巧”，全面掌握。

好啦，现在大家已经进入 F BASIC 的大门了。

是否没有感觉到……

第七节　本章小结

本章介绍了“程序、指令、语句、变量”等计算机术语的简单概念，并重点讨论了屏幕编辑的操作方法。

屏幕编辑操作在程序的输入和修改中用处颇大，希望大

家在这方面多下些功夫。

本章初步介绍了输出指令“PRINT”、内存程序列表指令“LIST”和程序运行指令“RUN”的基本使用方法，详细的总结将在下一章的小结中进行。

第二章
从“九九”歌开始

本章在学习“PRINT、LIST、RUN”三条指令的基础上，从讨论打印乘法“小九九”歌诀的程序编写方法入手，依次介绍：

循环语句说明指令	FOR—TO—STEP
	NEXT
无条件转向指令	GOTO
条件转向指令	IF…THEN
程序结束说明指令	END

第一节　电脑编写九九歌诀

早在上小学的时候，我们就熟悉了“一一得一、一二得二、二二得四……”的“九九”歌诀。

“九九”歌又叫“九九表”，俗称“小九九”，是进行乘、除法运算的基本口诀，只有朗诵在口、熟记在心才能正确地进行乘、除运算。通常，“九九表”是印在书上或文具盒等学习用具上，供小学生学习背诵。那么，能不能叫电脑打印出“九九”歌诀呢？请依照输入下面的程序（见程序 No. 2－1，“No. ”表示“第”几的意思。为了区分不同的程序，我们给每个程序取一个名字，以便称呼）：

No. 2－1

```
10 FOR I=1 TO 9 STEP1
20 PRINT I"X"I"="I*I
30 NEXT
```

键入“RUN”，回车后，将打印出如下结果

```
1×1=1
2×2=4
3×3=9
4×4=16
5×5=25
6×6=36
7×7=49
8×8=64
9×9=81
OK
■
```

以上结果虽然仅是九九表的一部分，但已能说明电脑可以打印出九九表。

那么，电脑是如何打印出九九表的呢？

这与循环语句有关。

第二节　循环语句

一、循环变量

细读程序 No. 2－1，字母“I”出现了五次，由第一章我们知道“I”叫做变量，在该程序 No. 2－1 中它还有个专用名字叫做“循环变量”，循环变量的作用马上就可以看到。

二、循环语句框架

程序 No. 2－1 中出现了“FOR、TO、STEP、NEXT”四条指令，这是新面孔，它们的作用是怎样的呢？

原来由这四条指令构成了循环语句的框架，其中

“FOR”指令是循环语句的起点，由它后面的数学表达式指出循环变量的开始数值（循环初值）；

“TO”指令引导出循环变量的结束值（循环终值）；

“STEP”指令定义循环变量“I”每一次循环所变化的幅度（循环步长，取循环迈一步的长度之意）；

“NEXT”指令控制一次循环结束，转入下一次循环。

三、循环语句的中文叙述

程序 No. 2－1 循环语句的中文叙述为：

令“I”的取值从“I＝1”开始，执行一次 20 语句的输出操作后“I”的值加一，继续执行 20 语句的输出操作，当“I＝9”的输出操作完毕时，结束循环。

四、循环语句的执行过程

显然，程序 No. 2－1 的执行过程就是：

I＝1　输出 1×1＝1，令 I＋1，I＝2

I＝2　输出 2×2＝4，令 I＋1，I＝3

I＝3　输出 3×3＝9，令 I＋1，I＝4

……

I＝9　输出 9×9＝81，令 I＋1，I＝10

已超过了循环终值“I＝9”的限制，停止执行输出语句，结束循环操作。

第三节　循环语句的应用

通过循环语句的学习，我们找到了解决反复运算问题的编程方法，请见以下几例。

一、平方、立方表

1～30之间整数的平方、立方表程序如下：

No. 2—2

```
10 FOR I=1 TO 30
20 PRINT I,I*I,I*I*I
30 NEXT
```

运行程序将打印出如下的结果：

```
1          1          1
2          4          8
3          9          27
4          16         64
.....
.....
29         841        24389
30         900        27000
OK
■
```

二、求阶乘

所谓“阶乘”就是对从1到某数之间(包括该数)的所有正整数求积，如“4”的阶乘等于1×2×3×4。一般某数的阶乘用“!”表示，如“4”的阶乘表示为“4!”。现在编写一个求“5!”的程

序，见程序。

No. 2—3

```
10 A=1
20 FOR B=1 TO 5
30 A=A*B
40 NEXT
50 PRINT A
```

运行程序将显示

```
120
OK
■
```

它表明

$5! = 120$

三、求前 N 项和

求自然数列前 100 项和的意思就是把 1～100 之间的所有自然数相加求和，通常记为

$$\sum_{n=1}^{100} N$$

实现程序见 No. 2—4。

No. 2—4

```
10 C=0
20 FOR D=1 TO 100
30 C=C+D
40 NEXT
50 PRINT C
```

运行程序，将显示

```
5050
```

```
OK
■
```

它表明自然数列前 100 项的和是 5050。

四、降序打印 20～2 之间的偶数

循环语句中的步长值也可以是负值，但这时循环变量的初值必须大于终值。降序打印 20～2 之间偶数的程序见程序 No. 2－5。

No. 2－5

```
10 FOR X=20 TO 2 STEP -2
20 PRINT X,
30 NEXT
```

运行程序将打印出如下结果：

```
20      18      16      14
12      10      8       6
4       2
OK
■
```

第四节　多重循环

所谓“多重循环”就是在循环语句之中再套以循环语句，犹如我们穿衣服一样，一层套一层。

一般以循环语句的层数命名循环的重数，如两层循环语句称为“双重循环”，三层循环语句称为“三重循环”等，相应地一层循环称为“单循环”。

一、完整的九九表

在第一节我们使用单循环打印出九九表的一部分，本节将使用双重循环打印出完全的九九表，实现程序见 No. 2－6。

No. 2－6

```
10 FOR I=1 TO 9
20 FOR J=1 TO I
30 PRINT J"X"I"="J*I
40 NEXT
50 NEXT
```

运行程序 No. 2－6，将打印出如下结果：

```
1×1=1
1×2=2
2×2=4
1×3=3
2×3=6
3×3=9
.....
9×9=81
OK
■
```

二、一屏显示的九九表

以上的结果确实是九九表的全部内容，但是输出的结果呈单列显示，要显示好几屏，不利于观察，若能显示在一屏中将方便使用，下面是改进后的程序。

No. 2－7

```
10 FOR I=1 TO 9
```

```
20 FOR J=1 TO I
30 PRINT J"X"I"="J*I;
40 NEXT
50 NEXT
```

运行程序将打印出如下结果：

```
1×1=1   1×2=2   2×2=4   1×3=3
2×3=6   3×3=9   1×4=4
2×4=8   3×4=12   4×4=16
.....
7×9=63   8×9=72   9×9=81
OK
■
```

现在，九九表的内容全部在一屏上显示出来。

可能有的朋友会问，程序 No. 2－6、No. 2－7 并无多大的差别，为什么输出结果有如此大的差异呢？仔细观察，差别仅在 30 语句行，程序 No. 2－7 中 30 行的最后多了一个分号（;），正是这个分号改变了输出语句的输出内容格式。它的作用是令各个输出项以紧凑格式显示。

三、整齐的九九表

程序 No. 2－7 虽然实现了一屏显示，但显示的内容较杂乱，仍不利于观察，程序 No. 2－8 将打印出整齐的九九表。

No. 2－8

```
10 FOR I=1 TO 9
20 FOR J=1 TO I
30 PRINT J"X"I"="J*I,
40 NEXT
50 NEXT
```

运行程序将打印出如下结果：

```
1×1=1          1×2=2
2×2=4          1×3=3
2×3=6          3×3=9
.....
.....
7×9=63         8×9=72
9×9=81
OK
■
```

结果按两列打印，整齐划一。这当然是30行语句的最后使用了逗号“,”所致，“,”的作用是令输出的内容分栏显示。

程序No.2－8已具有实用价值了，但美中不足的是由于打印“OK”和显示光标，结果最上面一行被移出屏幕。能否控制一屏显示既整齐又完整的结果呢，下一节将给出答案。

第五节　转向语句

电脑在执行程序时，一般是按行号的大小由小到大顺序执行的。但由于解决问题的需要，往往要求改变程序的执行顺序，转移到其它行号，这叫作程序的转向。程序转向由转向语句控制实现，转向语句有三种形式，分别为无条件转向、条件转向和开关转向。本章仅讨论前两种形式，开关转向将在下一章讨论。

一、无条件转向

无条件转向的控制指令是“GOTO”，它在英文中为“去往”的意思，“GOTO”指令后面添上转移的目标行号，就构成

了无条件转向语句，如

GOTO 100

GOTO 1500

等。

程序 No. 2－8 运行结果中的第一行之所以移出屏幕，就是因显示了程序运行结束提示“OK”和光标造成的。若能控制九九表打印完毕而程序运行不结束，就可不显示“OK”提示，也就可保证结果在一屏中显示。解决的方法是在程序的最后加入一行无条件转向语句

60 GOTO 60

。增加 60 行程序后，在打印九九表完毕，执行到 60 行时，程序在该行反复执行，永不结束（这种现象在计算机术语中叫做死循环），从而达到了不显示“OK”，显示完整九九表的目的。

朋友们可加入该条语句试试（注——在加入了 60 行程序并运行后，必须按“PAUSE”或“STOP”键才能打断程序的运行，退出死循环状态）。

二、条件转向

以上我们使用了双重循环语句打印出完整的九九表，下面再介绍用条件语句打印九九表的方法。

No. 2－9

```
10 I=1
20 J=1
30 PRINT J"X"I"="J*I,
40 J=J+1
50 IFJ<I+1 THEN 30
60 I=I+1
```

```
70 IFI<10 THEN 20
80 GOTO 80
```

运行程序 No. 2－9 将打印出与程序 No. 2－8 相同的结果。对比两个程序发现：程序 No. 2－9 比 No. 2－8 少了循环语句，多了“IF ……THEN……”语句，新出现的语句就是条件转向语句。条件转向语句的格式为：

IF＜关系式＞THEN＜行号或语句＞用中文叙述就是：

“如果”关系式成立“则”转移到指定的行号或执行其后的语句；不成立时顺序执行下一行语句。

程序 No. 2－9 中，50 行控制当被乘数“J”小于或等于乘数“I”时，转移到 30 行输出乘式和乘积；当被乘数大于乘数时顺序执行 60 行的语句。70 行控制乘数小于 10 时，转移到 20 行输出新乘数的乘式和乘积；当乘数等于 10 时结束输出，顺序执行 80 行的程序。

程序 No. 2－9 的工作过程是：由 40、50 行控制输出乘数取某个值时的全部算式，并限制被乘数不大于乘数；由 60、70 行控制改变乘数的取值，并限制乘数在 1～9 之间。

条件转向语句一般使用四种语句形式：

IF＜关系式＞THEN＜行号＞

IF＜关系式＞GOTO＜行号＞

IF＜关系式＞THENGOTO＜行号＞

IF＜关系式＞THEN＜语句＞

程序 No. 2－9 中使用了条件转向语句的第一种形式，第二、第三两种语句形式与第一种等价，读者可替换 No. 2－9 中的“THEN”指令验证；第四种形式表示在“THEN”指令后面还可以加其它语句，甚至再加一个条件转向语句等。

程序 No. 2－9 与 No. 2－8 相比稍嫌罗嗦，能不能化简一

下呢？请见程序 No. 2－10。

No. 2－10

```
10 I=1: J=1
20 PRINT J"X"I"="J*I,: J=J+1: IF J<I+1 THEN 20
30 I=I+1: IF I<10 THEN J=1: GOTO 20
40 GOTO 40
```

运行程序将打印出与程序 No. 2－8 相同的结果形式。

程序 No. 2－10 说明，BASIC 语言允许在一个语句标号下写入多个语句，语句之间用冒号隔开。

F BASIC 规定，一个语句标号下最多可容纳的字符数（包括标号、空格在内）为 256 个，也就是说，一个语句行的最大长度为 256。因此，只要程序的执行顺序允许，可以把多个带标号的语句紧缩为一个语句（程序 No. 2－10 中的 20、30 两行都包含了三条语句）。

现在用条件转移语句编写求自然数列前 100 项和的程序。

No. 2－11

```
10 A=0: B=0
20 A=A+1: B=B+A
30 IF A<100 THEN 20
40 PRINT "B="B
```

运行程序仍然打印程序 No. 2－4 相同的结果。

程序 No. 2－11 还可以写成 No. 2－12 的形式。

No. 2－12

```
10 A=B
20 A=A+1: B=B+A
30 IF A=100 THEN PRINT "B="B: END
40 GOTO 20
```

运行 No. 2—12 仍能打印出与 No. 2—4 相同的结果。

通过以上讨论可见，程序的编制是十分灵活的，一个问题可以有多种编程方法进行解决。

程序 No. 2—12 中又出现了一条新指令“END”。“END”在英文中是“结束”的意思，在 BASIC 程序中它是一条令程序运行结束的指令，当执行到该条语句时将无条件地停止。

第六节　指令与程序

通过两章的学习，我们对 F BASIC 语言已有了初步的了解，本节将对 F BASIC 的“指令、语句、变量”等有关概念进行系统的讨论。

一、指令

能够指挥计算机完成某种操作的符号（包括英文单词和英文字母组合）叫做指令。指令又叫做保留字或关键字，F BASIC 语言 BS. 2A 版本中的指令共分四大类：命令、语句、函数和运算符。

1. **命令**　命令为立即执行的指令，它是帮助用户对程序或计算机系统进行某种操作的一类词。如 LIST 可以对程序内容进行显示，RUN 可以运行程序等。命令不必写入程序行，电脑可以立即执行它所规定的操作。

2. **语句**　语句是处在程序中、能够完成某种操作或实现某项具体功能的一类词。如在暂缓执行型程序中 PRINT 指令就是语句。语句只有在程序运行时才执行其规定的操作。

3. **函数**　F BASIC 把求随机数、求绝对值等经常使用的数学函数预先编制了独立的程序供用户直接调用。用户调用

这种函数时，只要给出程序名并附以自变量很快就可得到计算结果。这种程序称为函数。F BASIC 语言不只包括数值函数，还包括用于处理文字的字符串函数及对数字与字符进行转换的转换函数。函数在程序中通常只能作为语句行的一部分使用。

4. **运算符** 运算符主要用来执行算术运算或者逻辑运算、进行关系比较或者对语句进行修饰说明。有些运算符直接取英文中的某个词，即与普通单词形式相同，如 AND、NOT 等；有的则用非字母符号表示，如“＊”“/”等。

二、指令的叙述方式

本书中介绍的指令一般采用如下的叙述方式：

1. **格式** 格式是指各个指令的使用方式。其中包括指令、参数、虚拟变量、必选项和可选项。有些指令同时具有两种以上的用途和用法时将分开列出。下面是格式中常用的几个符号。

“[]”——方括号中的内容为可选项，使用中可以选用也可以不用，但方括号不要写入语句中。

“()”——圆括号要写在语句中。

“<>”——尖括号表示语句体，语句中不要写。

“……”——省略号表示可选用若干个同类型的可选项。双引号要写在语句中。

2. **类型** 类型说明该指令属于命令、语句、函数中的哪一种或两种。

3. **简写** 简写表示电脑可以接受的指令缩写。每个指令的缩写后面都要跟一个圆点“.”(PRINT 的“?”缩写除外)，它是指令缩写的标识符。在使用指令缩写时一定要带圆点。如

“PRINT”可写为“P.”、“LIST”可写为“L.”等。

4. **功用** 功用系指某一指令在其特定格式下的功能及用途。该栏只对基本功能进行简要介绍，必要时还要介绍设立本条指令的背景和用意。具体功能在注释及说明中再随时解释。

5. **注释** 对格式中所涉及的各项参数、变量、必选项和可选项的含义、用法、限定等进行具体说明。同时还对词条的来源和英文含义进行解释。

6. **示例** 示例用于简要说明指令的使用格式或功能。

7. **说明** 主要用来对指令的功能与作用，作更进一步的补充，指出注意事项等。

三、语句

语句有狭义、广义之分，狭义概念的语句是指处于程序中的保留字；广义概念的语句则是指能完成某项操作的字符串，它通常包括三部分：标号（行号）、语句定义符和语句体（数学或逻辑表达式等）。

1. **标号** 每个语句前都要有一个数字，这个数字叫做行号或语句标号。行号是无符号整数，取值范围为0～65534，不要求连续（一般上下行间都留出一定的间隔以方便程序修改，补插新的语句行）。各语句输入电脑时不一定按行号的大小顺序输入，行号杂乱的语句输入电脑后，电脑会自动按行号的大小顺序排列整理好。

2. **语句定义符** 语句定义符就是计算机指令。它规定计算机执行一定的操作。

3. **语句体** 语句体是跟在语句定义符后面的、需要去执行的具体内容。

每个语句不一定都有这三部分，但行号和语句定义符是

一定要有的。

四、常量与变量

1. **常量** 在程序的执行过程中，数值始终保持不变的量称为常量。

F BASIC 中常量的取值范围为－32768～32767。

2. **变量** 在程序的运行中可以改变的量称为变量。

F BASIC 中有两类变量：简单变量和下标变量。简单变量又分整型数变量和字符串变量两种。一般把整型数变量简称为整型变量，把字符串变量称为串型变量或串变量。下标变量的形式为

A(3)　　A$(4)

等，也有数型、串型之分，我们将在“DIM”指令的讨论中详细介绍。

变量名的构成必须遵守以下规则：

(1)变量名可以用一个、两个、三个……字符表示，最后一个字符说明变量的类型。“$”表示串变量，没有“$”的为数值变量。

(2)变量名的第一个字母必须是英文大写字母，以后的可以是任意字母或数字。

(3)保留字不能用作变量名或变量名的一部分。

(4)变量名最多可以取 255 个字符，电脑会按照你命名的变量名称存储和打印赋予变量的值，但运算时它只识别除“$”号外的前两个字符。如“ABC、ABD”两个变量，电脑当做一个变量“AB”进行处理。然而“ABC、ABD$”则不同，“ABC”是数值变量，“ABD$”是串变量。

第七节　本章小结

一、已学习的指令综述

以上我们共讨论了 PRINT LIST RUN FOR－TO－STEPNEXT GOTO IF THEN END 十一条指令的使用方法，现总结如下供查询使用。

1. 输出指令(PRINT)

格式：PRINT＜数据＞

类型：语句或命令

简写：P. 或?

功用：显示数值变量或串变量的内容。

注释：PRINT 在英文中为“印刷”的意思。

(1)数据是指被显示的内容，它可以是数值型的，也可以是字符串型的。无论数型还是串型，既可取常数形式，也可以取变量形式，还可以取表达式形式。取常数形式时直接显示该常数；取变量形式时则把该变量的当前值显示出来。表达式是指用常数或变量和运算符组成的算式，当输出表达式时，首先计算表达式的值，然后再显示该数值。

(2)PRINT 可以输出多项数据。输出的数据之间可以用逗号或分号进行分隔。逗号和分号的作用是不同的。逗号可使数据各项按分区方式显示(所谓“分区”就是空隔 4 个或 8 个空格显示下一输出内容)，如程序 No. 2－8～No. 2－10 中由于使用了逗号，输出的九九表分两栏显示；分号的作用与输出项之间不加符号相同，它使数据按紧凑格式显示，如程序 No. 2－7。但 PRINT 指令的最后输出项无逗号或分号时将换

行显示，如程序 No. 2－6。

说明：PRINT 指令有两个身分（类型），作为语句身分时可以放在语句中使用（带标号的程序）；作为命令身分时可以放在命令行中使用（不带标号的程序）。FBASIC 中有许多指令具有双重身分，以后的介绍中将在"类型"一栏注明。

2. **循环语句（FOR TO STEP NEXT）**

格式：FOR I＝n TO m STEP s

NEXT

类型：语句或命令

简写：F. I＝n TO m ST. s

N.

功用：FOR TO STEP 为循环的起始语句，它可以确定循环变量"I"的初值、终值、步长值；NEXT 为循环的终止语句，它标志着循环体的结束（循环起始语句和终止语句之间的语句称为循环体）。

注释：

I－循环控制变量，可用任意简单变量表示。

n－循环控制变量初值。

m－循环控制变量终值。

s－循环变量每次循环的增量，也叫循环步长。步长为 1 时可以省略 STEP。

FOR TO STEP NEXT 的意思是：循环从变量等于初值开始，到变量超出终值时结束，每循环一次变量增加一个步长值。

说明：

（1）循环的初值、终值、步长值一般为数值，也可以取变量形式。

(2)循环语句可以多层嵌套,只要内存允许,最多可以嵌套九重。

(3)在循环语句的编程中,允许从内循环跳到外循环或跳出循环体之外,但不允许从循环体外部跳入循环体之内。这一点在编程中务必充分注意。

3. 无条件转移语句 GOTO

格式:GOTO<行号>

类型:语句

简写:G.

功用:改变程序顺序执行的规则,转向规定的行号继续执行。

注释:GOTO 在 BASIC 语言中是"转向或转去执行"的意思。

说明:GOTO 语句中的行号为转向的目的地,代表所要转向的程序行,行号要取常数。运行中当程序执行到这一 GOTO 语句时,将无条件地转移到规定的行号去,并从这一行号起继续执行。

4. 条件转向语句(IF THEN)

格式:IF<关系式>THEN<行号>

IF<关系式>GOTO<行号>

IF<关系式>THEN GOTO<行号>

IF<关系式>THEN<语句>

类型:语句

简写:IF<关系式> T.<行号>

IF<关系式> G.<行号>

IF<关系式> T.G.<行号>

IF<关系式> T.<语句>

功用:根据关系式成立与否决定程序是否转移和转移的目的地。

IFTHEN 语句也叫 IF 语句,它可以改变程序顺序执行的规则。

注释:IF……THEN……为"如果(条件成立)则……"的意思。

说明:关系式有两种,关系表达式和逻辑表达式。关系表达式是由小于号(<)、大于号(>)、等于号(=)、不等于号(<>)、大于等于号(>=)、小于等于号(<=)等关系运算符连接起来的式子。逻辑表达式是由"与"(AND)、"或"(OR)、"非"(NOT)、"异或"(XOR)逻辑运算符连接起来的式子。

另外,关系式还可以取关系表达式和逻辑表达式混合组成的形式。

如果关系式成立(即经运算后结果为真)则表明 IFTHEN 语句判断的条件成立,BASIC 将会转移到 THEN 或 GOTO 后的行号继续执行或执行 THEN 后面的语句。若关系式不成立,BASIC 就跳过本语句,执行后面的语句,此时相当于 IF THEN 语句行不存在。

IF 语句中,关系式后面可以用 THEN、GOTO 或 THENGOTO,三种形式效果一样,因此前三种格式可以任选。

5. **结束语句(END)**

格式:END

类型:语句

简写:E.

功用:终止程序执行。

注释:END 为结束的意思。

说明:END 指令单独组成语句,插在程序行的中间,控制终止程序的运行。若 END 放在程序的最后可以省略。

6. **程序列表指令(LIST)**

格式:LIST [【m】—【n】]

类型:命令、语句

简写:L.

功用:显示部分或全部程序语句内容,如果程序语句不是按行号顺序编辑的,LIST 命令列表时可将程序按行号顺序整理后再显示出来。

当我们编辑一个程序或要了解程序内容时,就需要使用这条语句。

注释:LIST 在英文中为清单、列表的意思。

说明:m、n 为任选项,它们的选用和对应的功能如下:

LIST m	列出行号为 m 的一行程序
LIST m—	列出 m 行及 m 行以后的所有程序
LIST —n	列出起始行至 n 行的所有程序
LIST m—n	列出 m 行至 n 行的程序
LIST	列出全部程序

示例:

LIST 30	仅列出 30 行的程序
LIST 50—	列出 50 行至程序尾的所有程序
LIST —100	列出从程序的第一行开始至 100 行的程序
LIST 20—100	列出 20～100 行的所有程序

语句中的“—”号也可以用“,”代替,两者的作用相同。

7. **程序运行指令(RUN)**

格式:RUN [m]

类型:命令、语句

简写:R.

功用:运行内存中的程序。

注释:RUN 在英文中为跑、运行的意思。

如果在 RUN 后跟有行号 m,程序将从该行号所指定的语句行开始向下运行。若没有行号则从程序的第一行开始执行。

示例:RUN 50

说明:RUN 命令执行时,首先清除内存中现有程序中的变量。执行一次 RUN 后,程序并没有破坏,可以多次执行。

二、运算符号、表达式、记号

正确地使用各种运算符号,是编程中少走弯路的要点之一,现将 F BASIC 的运算符号和表达式的叙述方式汇总如下,供编程中查阅。

运算符号分算术运算符号、关系运算符号和逻辑运算符号。

1. **算术运算符号** 算术运算符号共有五种,它们的表达形式和规则见表 2－1。

表 2－1 算术运算符号与表达式

运算符号	运算内容	编程示例	数学表达式
+	加	A+B	A+B
－	减	A－B	A－B
*	乘	A * B	A×B
/	除	A/B	A÷B 或 A/B B≠0
MOD	求余数	A MOD B	A÷B 后余数

(1)MOD 为计算“除”运算时的余数。如：

18÷4＝4……2

则有　18MOD4＝2

(2)运算优先顺序

在数学的运算中，有先乘除后加减、先括号内后括号外的运算优先顺序(也叫优先级)，在计算机编程中仍遵循这一原则。各算术运算符号的优先级别如下：

①“()”括号

②“＊”、“/”

③MOD

④“＋”、“－”

在同一算式中，若两个运算符号的优先级相同，则遵循先左后右的原则。

若要改变运算的优先顺序，应将先运算的内容用括号“()”表示，如表 2－2。

表 2－2　数学表达式的编程表达方式

数学中的算术表达式	编程中的算术表达式
AX＋Y	A＊X＋Y
$\frac{X+Y}{2}$	(X＋Y)/2
A[X＋B(X＋C)]	A＊(X＋B＊(X＋C))
$\frac{A+1}{3B}\div\frac{C-3}{4D}$	((A＋1)/(3＊B))/((C－3)/(4＊D))

F BASIC 编程中没有大、中、小括号之分，一律使用圆括

号(小括号),编程中要注意使用的左、右圆括号的成对关系。

2. **关系运算符号** 关系运算符号用于两个常量或变量的比较,比较后给出一个数值标志,若比较结果满足条件时其值为"－1",否则其值为"0"。

F BASIC 的关系运算符号见表 2－3。

表 2－3 关系运算符号与表达式

运算符号	运算内容	示 例	数学表达式
=	两边相等	A=B	A=B
<>	两边不等	A<>B	A≠B
>	左边大于右边	A>B	A>B
<	左边小于右边	A<B	A<B
>=	左边大于或等于右边	A>=B	A≥B
<=	左边小于或等于右边	A<=B	A≤B

在 F BASIC 程序中,关系运算符号可用作条件判断语句,如:

IF X>0 THEN 100

其含义是:如果 X 的值大于零,则转向到程序标号为 100 的语句行执行。

3. **逻辑运算符号**

本部分内容涉及到二进制数的概念,不熟悉二进制数的读者可参阅丛书之六《汇编语言》。也可先跳过本部分内容。

逻辑运算符号主要用于二进制数"0"及"1"的逻辑运算,其含义如表 2－4 所示。

表 2—4　逻辑运算符号与表达式

运算符号	运用内容	示　例	逻辑表达式
NOT	否定(非)	NOT A	
AND	逻辑积(与)	A AND B	A∧B
OR	逻辑和(或)	A OR B	A∨B
XOR	逻辑排它和(异或)	A XOR B	

(1)逻辑运算优先顺序

① NOT

② AND

③ OR

④ XOR

(2)逻辑运算真值表

NOT

A	NOT
0	1
1	0

AND

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

OR

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

XOR

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

注——变量与逻辑运算符号连续书写时，中间要留有间隔。

示例：

① NOT 13

13 =00001101B(后缀 B 表示二进制数)

NOT 13 =11110010B(按位取反)

=－14(取其补码)

② 50 AND 5

50 =00110010B

5=00000101B

50 AND 5 =00000000B(按位相与)

=0

③ 17 OR 8

17 =00010001B

8 =00001000B

17 OR 8 =00011001B(按位相或)

=25

④ 11 XOR 12

11 =00001011B

12 =00001100B

12 XOR 11=00000111B(按位相异或)

=7

以上示例请对照真值表分析。

4. **运算规则** 当三种运算符号在一个表达式中同时出现时，将遵循如下运算规则：先计算括号内，然后依次是函数、乘除、MOD、加减、关系运算、逻辑运算。它们的运算优先级为：

① 以圆括号包括的部分

② 函数

③ *、/(乘、除)

④ MOD(求余运算)

⑤ +、-(加、减)

⑥ =、<>、>、<、>=、<=

⑦ NOT(否定)

⑧ AND(逻辑积)

⑨ OR(逻辑和)

⑩ XOR(逻辑异或)

5. **记号及特殊用途** F BASIC 语言加强了记号(符号)的使用,从而大大压缩了 F BASIC 程序的书写空间,使较小的内存空间发挥较大的作用。各记号的意义如表 2-5。

表 2-5 F BASIC 程序记号用途对照表

名称	记号	含义	示例
连号	-	LIST 指令中,程序可能超过 24 行,使用连号控制程序分段列出	LIST-9 LIST 100-300 LIST 310-
逗号	,	在 PRINT 等指令中,当多于一个操作数时,每个数之间用逗号分开,使电脑接受并继续操作。	PRINT A,B,C INPUT A,B,C READ A,B,C,D DATA 10,13,5,6,8
冒号	:	在同一程序行中有两条以上语句时使用。	A=B+C:PRINT A,B,C
分号	;	在 PRINT 等指令中作多项操作时,项之间用分号隔开,以使电脑了解执行的先后。	PRINT "ABCD";A INPUT"A=";A
问号	?	用"?"代替 PRINT 以提高输入速度。	?A 与 PRINTA 相同
其它	&H '	十六进制数标志 用"'"代替 REM 指令	&H2C 即十进制数 44' 的作用与 REM 相同

第三章
梁山好汉排座次

排序——是计算机编程中的一项重要内容，也是电脑的特长之一。本章结合排序程序的编写，循序介绍：

键盘扫描指令	INKEY $
键盘输入指令	INPUT
清屏指令	CLS
清除内存指令	NEW
数组说明指令	DIM
两变量内容交换指令	SWAP
读数指令	READ
置数指令	DATA
数据区恢复指令	RESTORE
注释指令	REM
子程序调用指令	GOSUB
子程序返回指令	RETURN

第一节　显示按键上的字符

用 BASIC 语言编写一个简单的程序也可以进行键盘练习。请输入程序 No. 3－1 试试。

No. 3－1

```
10 A$ =INKEY$ (0)
```

```
20 PRINT A$"";
30 GOTO 10
```

运行程序，在程序行下的行首显示着闪烁的光标，等待你按键。这时你按哪一个键，哪一个键上的字符将显示在屏幕上，显示的字符之间相隔一个空格。现在，No. 3－1 就是一个键盘练习程序，可以让朋友当老师提问字符，你按键盘回答，考核你的击键速度和键盘熟悉程度。

程序 No. 3－1 使用了一条新指令“INKEY $”，“INKEY $”为键盘扫描指令，它的功能就是对键盘进行扫描，等待读取输入的某个字符，一旦读到字符就把该字符赋值给字符串变量 A $，由 20 行输出 A $ 中的内容。A $ 后用引号括起的空格的作用，是为了使屏幕上显示的字符之间产生间隔。

第二节　输入一串字符

一、输入你的姓名

程序 No. 3－1 一次只能输入一个字符，能否一次输入多个字符呢？请输入程序 No. 3－2。

No. 3－2

```
10 INPUT A$
20 PRINT A$"";
30 GOTO 10
```

运行程序，将在程序行下的行首显示“?”，后跟闪烁的光标。这时依次按字母键，如输入“李明”的汉语拼音“LI MING”，将在“?”后显示“LI MING”，回车后将在下一行显示“LI MING”，其后面空一格显示“?”和闪烁的光标，等待输入

下一串字符。

```
RUN
? LI MING
LI MING? ■
```

程序 No. 3－2 中使用了一个新指令“INPUT”，“INPUT”指令的功用是接受键盘输入的数据，并把该数据赋给其后的变量。

二、一次输入多个数据

“INPUT”指令可以依次读入多个数据，并把数据依次赋给多个变量，同时还可以把有关提示内容写入“INPUT”语句体，在等待用户输入数据时进行提示（见程序 No. 3－3）。

No. 3－3

```
10 INPUT "A,B,C=";A,B,C
20 D=A+B: E=A-B: F=A*B
30 PRINT "A+B="D, "A-B="E, "A*C="F
```

运行程序，将显示

A,B,C=? ■

闪烁的光标等待输入“A,B,C”三个数据，依次输入

5,6,7↙

后，屏幕上将显示

```
A,B,C=? 5,6,7
A+B=11      A-B=-1      A*C=35
OK
■
```

对于“INPUT”指令后要求输入多个数据的情况，多个数据可以一次输入完毕后按回车键（各个数据之间用逗号隔开），也可以每输入一个数据按一次回车键（这时数据之间不

用输入逗号),连续输入多次。

三、重复输入数据

在程序 No. 3－3 中若再添一行

40 GOTO 10

语句,运行程序,将显示

```
A,B,C=? 5↙
? 6↙
? 7↙
A+B=11      A-B=-1      A*C=35
A,B,C=? 8↙
? 9↙
? 10↙
A+B=17      A-B=-1      A*C=80
A,B,C=? ■
```

闪烁的光标等待用户继续输入数据,键盘输入和计算输出可一直持续下去,直到按“Pause”或“STOP”键打断程序的运行。

可见,使用“INPUT”语句可以很方便地对确定算式的许多组不同数据进行反复输入运算,如计算第一章中李明、张华的平均学习成绩,可编写如 No. 3－4 的程序。

No. 3－4

```
10 INPUT "XING MING, SHU XUE, YU WEN="; A$,A,B
20 C=A+B: D=C/2
30 PRINT A$
40 PRINT "ZONG CHENG JI="C
50 PRINT "PING JUN CHENG JI="D
60 PRINT
```

```
70 GOTO 10
```

第 10 行引号中的内容依次为“姓名、数学、语文”键盘输入内容提示。

RUN↙(表示键入运行程序命令按回车键,即运行程序,下同)屏幕显示

```
XING MING,SHU XUE,YU WEN=? ■
```

依次输入李明的姓名“LI MING”、数学成绩“96”分、语文成绩“88”分,回车后,屏幕显示

```
XING MING,SHU XUE,YU WEN=? LI MING,96,88↙
LI MING
ZONG CHENG JI=184
PING JUN CHENG JI=92
XING MING,SHU XUE,YU WEN=? ■
```

依次输入张华的姓名“ZHANG HUA”、数学成绩“93”分、语文成绩“85”分,回车后,屏幕显示

```
XING MING,SHU XUE,YU WEN=? ZHANG HUA,93,85↙
ZHANG HUA
ZONG CHENG JI=178
PING JUN CHENG JI=89XING MING,SHU XUE,YU WEN
=? ■
```

显然,程序 No. 3—4 可以连续计算多个学生的总成绩和平均成绩。

四、清屏语句 CLS

通过以上诸个程序的运行,大家可能已对程序运行时,屏幕上原来显示的内容与程序运行中新显示的内容掺杂在一起感到观察不便,希望在运行程序时把屏幕上原来显示的内容清除掉。使用清屏命令“CLS”就可以达到目的。

现在，在程序 No. 3－4 中加入

5 CLS

再运行程序，将首先把屏幕清除干净，在屏幕的第一行显示：

XING MING,SHU XUE,YU WEN=? ■

以后的操作同上面一样，不多赘述。

五、程序清除命令 NEW

在学习阶段，演示的程序既小且多，当输入一个新程序时，新输入的语句行可以覆盖原来对应的程序行，这固然不须多虑。但没有对应语句的原来程序行将继续保留在内存中，这将破坏新程序的完整性，并可能造成新程序运行错误。如在程序 No. 3－4 后再输入新程序 No. 3－5，看一看将会产生什么样的结果。

No. 3－5

```
10 INPUT "BEI CHENG SHU,CHENG SHU=";A,B
20 PRINT A,B,A * B
```

程序 No. 3－5 的功能是从键盘读入被乘数、乘数，然后分别打印出被乘数、乘数和乘积，但运行程序，依次输入被乘数 33、乘数 55 后，屏幕显示的结果将如下所示。

```
BEI CHENG SHU,CHENG SHU=? 33,55
33      55      1815
ZONG CHENG JI=0
PING JUZ CHENG JI=0
BEI CHENG SHU,CHENG SHU=? ■
```

程序 No. 3－5 的运行结果的后四行即是程序 No. 3－4 的影响所致。这时若列出程序清单，将看到程序 No. 3－4 的 30～70 行的内容依然存在。

脑筋灵活的朋友可能已经想到在程序 No. 3－5 中加入

30 END

就可控制不发生上述错误。但这种方法虽然有效却功能有限，因为它只能隔断“END”语句行以后的程序干扰，但对于其以前的语句行将无能为力。若程序 No. 3－4 中有一个 15 行或 25 行，“END”语句将失去作用。因此，最有效的方法就是输入新程序前，把内存中原来的程序清除干净。清除内存的指令是“NEW”，在英文中是“新”的意思，即把内存置为新的状态。

希望大家养成在编写新程序前先执行一次“NEW”命令的习惯。

第三节　键盘输入成绩排名次

一、数组及数组说明语句

根据成绩排列名次在数学中叫做“排序”。

在排序程序的编写中，数组和下标变量是必不可少的，下面我们来学习这方面的知识。

1. **下标变量**　顾名思义，下标变量就是带有下标的变量，如 A1、A2、A3、A11、A123 等，它们虽然使用相同的变量名“A”，但由于使用了不同的下标“1、2、3、11、123”，所以应视为 5 个不同的变量。为了在编程中与简单变量 A1、A2 相区别，计算机语言中，一般把上述下标变量记为 A(1)、A(2)、A(3)、A(11)等。使用下标变量的目的是为了使一些性质相近的变量便于归类。如对 A(K)而言，K＝1 时 A(K)就是 A(1)，K＝2 时 A(K)就是 A(2)……。

F BASIC 语言允许下标变量名中包含两个下标，其形式

为 A(I,J)。我们称一个下标的为单下标变量;两个下标的为双下标变量。

2. **数组** 由同一个变量名组成的下标变量的集合叫数组,如上面讨论的 A(1)、A(2)、A(3)……就是以"A"为变量名的数组,可简称"A 数组"或"数组 A"。有了数组,我们就可以把性质相近的数据放入一个数组下的各个下标变量中,这样不仅便于区分变量的种类(仅看下标变量名就可区分),而且对变量赋值或从变量中读数都十分方便(只要调整下标变量的下标就可改变变量)。

通常,对由单下标变量组成的数组称为一维数组;由双下标变量组成的数组称二维数组。

3. **数组说明语句** 数组在使用前要先进行定义,使用"DIM"指令就可以定义数组。定义数组的目的就是规定数组的名称及所包含下标变量的个数。一个指令可以定义若干个数组,每个数组名之间用逗号隔开,如

DIMA(5),B(6),C(7),A$(4)等。

二、期终考试排名次

现有十名学生的期终考试成绩如表 3-1 所示,编写程序按分数的高低排出名次。

为方便输入,我们仅以学生的学号为标记。成绩排序程序见 No. 3-6。

```
No. 3-6
10 CLS: DIM A(10),D(10)
20 FOR I=1 TO 10
30 INPUT "XUE HAO, SHU XUE,YU WEN=";A,B,C
40 A(I)=A: D(I)=B+C
```

```
50 NEXT
60 FOR K=1 TO 9
70 FOR J=K+1 TO 10
80 IF D(J)>D(K) THEN X=A(J): A(J)=A(K): A(K)=X: Y
   =D(J): D(J)=D(K): D(K)=Y
90 NEXT
100 NEXT
110 CLS
120 FOR I=1 TO 10
130 PRINT "DI";I;"MING";A(I);D(I)
140 NEXT
```

表 3—1　期终考试成绩一览表

姓　名	学　号	数　学	语　文
李　明	1000	89	83
张　华	1001	97	82
高海燕	1002	67	93
郭放声	1003	84	87
宋海波	1004	99	86
谭永民	1005	71	96
邱吉祥	1006	94	78
何明初	1007	86	89
李育行	1008	77	96
孙明远	1009	100	98

程序说明：

10 行定义了两个数组 A、D，分别用于存放 10 名学生的学号和总成绩。

20～50 行控制从键盘读入各学生的学号、数学、语文成绩，把学号放入数组“A”的各下标变量中，把成绩之和放入数组“D”的各下标变量中。

60～100 行使用了双重循环，按总成绩的高低进行排序，使成绩高的排在前面。

110～140 行输出排好名次的结果。

现在，我们来考察程序 No. 3－6 的功能。

运行程序，清屏，显示

```
XUE HAO,SHU XUE,YU WEN=? 1000,89,83↙
XUE HAO,SHU XUE,YU WEN=? 1001,97,82↙
XUE HAO,SHU XUE,YU WEN=? 1002,67,93↙
……
……
XUE HAO,SHU XUE,YU WEN=? 1009,100,98↙
```

依次输入十个学生的学号、数学、语文成绩回车后，将清屏，打印出排好名次的结果。

```
DI 1 MING 1009 198
DI 2 MING 1004 185
DI 3 MING 1001 179
DI 4 MING 1007 175
DI 5 MING 1008 173
DI 6 MING 1006 172
DI 7 MING 1000 172
DI 8 MING 1003 171
DI 9 MING 1005 167
DI 10 MING 1002 160
OK
■
```

程序 No. 3－6 中未考虑并列名次的情况。

三、变量内容交换指令

程序 No. 3－6 的 80 行中，数组 A、D 的两个变量内容的

交换是通过中间变量“X、Y”实现的，每一个交换要使用三条语句。其实 F BASIC 专门提供了一个变量内容交换指令“SWAP”，两个变量的内容交换时，只要把两个变量名之间用逗号隔开，依次写在“SWAP”指令后面就可以了。如

```
SWAP A(J),A(K)
SWAP D(J),D(K)
```

显然，使用“SWAP”指令可使变量交换程序得到简化。请拿以上的语句去替换程序 No. 3－6 中 80 行 THEN 后面的变量内容交换语句，运行程序将输出相同的结果。

程序 No. 3－6 只对学生的总成绩进行排名次，能否分别按数学、语文的成绩进行排序，以检查数学、语文的名次情况呢？应该可以，只须对程序 No. 3－6 稍加修改就可以实现。这一问题，作为练习留给读者完成。

第四节　程序输入成绩排名次

程序 No. 3－6 中的学生成绩必须在程序运行后才能输入，每运行一次要输入一次，发现输入错误后也只能听之认之，使用极其不便。能不能把成绩等数据像编写程序那样直接写在程序中，一次输入多次使用呢？

这是完全可以的。

置数指令“DATA”和读数指令“READ”提供了实现的手段。在英文中“DATA”为“数据”的意思，“READ”为“读”的意思，两条指令互为对应，“DATA”指令就是为读数指令“READ”提供数据的。程序运行中，执行到 READ 指令时，READ 指令后的各个变量会依次从 DATA 语句中读取数据，见如下程序段：

```
10 READ A,B,C,D
20 DATA 25,35,45,55
30 PRINT A,B,C,D
```

执行程序后，将打印

25　　35　　45　　55

由于 DATA 语句是非执行语句，因此它可以写在程序中的任意位置，如上述程序改为

```
10 DATA 25,35
20 READ A,B,C,D
30 PRINT A,B,C,D
40 DATA 45,55
```

仍能打印相同的结果。

现在使用这两条指令，把学生成绩排名次的程序改写如程序 No. 3－7。

No. 3－7

```
10 CLS: DIM A(10),D(10)
20 FOR I=1 TO 10
30 READ A,B,C
40 A(I)=A: D(I)=B+C
50 NEXT
60 FOR K=1 TO 9
70 FOR J=K+1 TO 10
80 IF D(J)>D(K) THEN SWAP A(J),A(K):SWAP D(J), D
   (K)
90 NEXT
100 NEXT
110 CLS
120 FOR I=1 TO 10
130 PRINT "DI";I;"MING";A(I);D(I)
```

```
140 NEXT
150 DATA 1000,89,83,1001,97,82,1002,67,93,1003,84,87,
    1004,99,86,1005,71,96,1006,94,78,1007,86,89,1008,77,
    96,1009,100,98
```

程序 No. 3－7 与 No. 3－6 的区别在于，30 行用“READ”指令代替了“INPUT”指令，多出了写入成绩数据的 150 行。

运行程序将打印出与 No. 3－6 相同的结果。

显然，程序 No. 3－7 可以对更多的学生排名次，修改方法是根据学生的人数，改变 10 行数组的下标数值和 20、60、70、120 行的循环终值以及添加 DATA 语句行。

第五节　多科成绩排名次

第三节曾留给读者编写以各科成绩排序的练习题，现在带领大家做一个数学成绩、语文成绩、总成绩分别排序的编程练习。我们分编程分析、编写程序、程序说明、运行程序四段进行介绍。

一、编程分析

程序 No. 3－7 中，数组 A 用于存放学生的学号，数组 D 用于存放总成绩，两个数组记录了一项排序数据。现在新增加的数学、语文两项排序，只要再增加四个数组，使用相同的程序就可实现。我们定义数组 B 存放数学成绩对应的学号，定义数组 E 存放数学成绩；定义数组 C 存放语文成绩的学号，定义数组 F 存放语文成绩。只要把程序 No. 3－7 重复三次就可编写出多科成绩的排序程序。

二、编写程序

程序见 No. 3－8。

No. 3－8

```
10 CLS: DIM A(10), D(10)
20 FOR I=1 TO 10: READ A,B,C: A(I)=A: D(I)=B+C:
   NEXT
30 FOR K=1 TO 9: FOR J=K+1 TO 10
40 IF D(J)>D(K) THEN SWAP A(J), A(K): SWAP D(J), D
   (K)
50 NEXT: NEXT
60 CLS: PRINT "ZONG CHENG JI PAI XU": PRINT
70 FOR I=1 TO 10: PRINT "DI";I;"MING";A(I);D(I):NEXT
80 DATA 1000,89,83,1001,97,82,1002,67,93,1003,84,87,
   1004,99,86,1005,71,96,1006,94,78,1007,86,89,1008,77,
   96,1009,100,98
100 A$=INKEY$(0)
110 RESTORE
120 CLS: DIM B(10), E(10)
130 FOR I=1 TO 10: READ A,B,C: B(I)=A: E(I)=B:
    NEXT
140 FOR K=1 TO 9: FOR J=K+1 TO 10
150 IF E(J)>E(K) THEN SWAP B(J),B(K): SWAP E(J), E
    (K)
160 NEXT: NEXT
170 CLS: PRINT "SHU XUE CHENG JI PAI XU"
180 FOR I=1 TO 10: PRINT "DI";I;"MING"; B(I); E(I):
    NEXT
200 A$=INKEY$(0)
```

```
210 RESTORE
220 CLS: DIM C(10),F(10)
230 FOR I=1 TO 10: READ A,B,C: C(I)=A: F(I)=C:
    NEXT
240 FOR K=1 TO 9: FOR J=K+1 TO 10
250 IF F(J)>F(K) THEN SWAP C(J),C(K): SWAP F(J), F
    (K)
260 NEXT: NEXT
270 CLS:PRINT "YU WEN CHENG JI PAI XU"
280 FOR I=1 TO 10: PRINT "DI";I;"MING";C(I);F(I):
    NEXT
```

三、程序说明

10～80 行为程序 No. 3－7 的缩写，是按总成绩排序的处理程序。其中新添的 60 行用于说明排序种类(按总成绩排序)。

100 行的读键语句是控制程序执行暂停，等待用户按任意键后再向下执行程序。

110 行“RESTORE”指令的作用是“恢复数据区指针”。因为 80 行的数据在 20 行读数结束后，数据指针已经指向数据区的尾部，若继续读数据只有把 80 行的数据复制一行(如 90 行)，才能有数据供 130 行的 READ 语句读取。由于数据是相同的，复制一行不仅增大了程序量，多占用了内存，而且程序也不简洁。使用“RESTORE”指令后，系统的数据指针就恢复到 80 行的第一个数据处，当执行 130 行的 READ 语句时又可以从头读取数据。

120～180 行的作用是按数学成绩排序和输出。

200～280 行的作用是按语文成绩排序和输出，其中的

200、210 行的作用与 100、110 行相同。

四、运行程序

运行程序 No. 3－8，将清屏显示按总成绩排序的结果如下：

```
ZONG CHENG JI PAI XU
DI 1 MING 1009 198
DI 2 MING 1004 185
DI 3 MING 1001 179
DI 4 MING 1007 175
DI 5 MING 1008 173
DI 6 MING 1006 172
DI 7 MING 1000 172
DI 8 MING 1003 171
DI 9 MING 1005 167
DI 10 MING 1002 160
```

运行结果与程序 No. 3－7 相同。

按任意键后，清屏，将显示单按数学成绩排序的结果如下：

```
SHU XUE CHENG JI PAI XU
DI 1 MING 1009 100
DI 2 MING 1004 99
DI 3 MING 1001 97
DI 4 MING 1006 94
DI 5 MING 1000 89
DI 6 MING 1007 86
DI 7 MING 1003 84
DI 8 MING 1008 77
DI 9 MING 1005 71
```

DI 10 MING 1002 67

按任意键后，清屏，将显示单按语文成绩排序的结果如下：

YU WEN CHENG JI PAI XU

DI 1 MING 1009 98

DI 2 MING 1008 96

DI 3 MING 1005 96

DI 4 MING 1002 93

DI 5 MING 1007 89

DI 6 MING 1003 87

DI 7 MING 1004 86

DI 8 MING 1000 83

DI 9 MING 1001 82

DI 10 MING 1006 78

五、注释语句

No. 3－8 以前的程序均较短小，程序结构一目了然，较容易读懂，但要读懂 No. 3－8 就要琢磨一番，若遇到更大的程序，读起来困难就更大。F BASIC 专门提供了一条注释指令“REM”，使用它可以在程序中编写注释语句，如

10 REM This is main program.

……

80 REM pai xu cheng xu　等等。

由“REM”指令引导的语句为非执行语句，它仅起到程序的注释作用，不影响程序的执行。因此，如果需要可在每行程序尾加一条注释语句，以便于程序的阅读和交流。

我们可以在程序 No. 3－8 中加入如下注释语句：

5 REM An zong cheng ji pai ming ci.

95 REM An shu xue cheng ji pai ming ci.

195 REM An yu wen cheng ji pai ming ci.

三行注释语句的内容分别为“按总成绩排名次、按数学成绩排名次、按语文成绩排名次”。有了这三行语句，阅读程序No. 3－8就容易多了。

第六节　排序程序的优化

程序No. 3－8虽然实现了按总成绩、数学成绩、语文成绩分别排序的目的，但程序的重复部分太多，太不经济。本节将要讨论的子程序调用指令“GOSUB、RETURN”，会使程序变得紧凑精练。

1. **子程序的概念**　在计算机编程中，通常把那些功能相同、且反复使用的程序段归纳为一个独立的程序，放在主程序之外，供各段程序调用。这种独立的程序段就称为子程序。

子程序可以由一行或数行语句构成，其第一行程序的行号即为该子程序的入口，其它程序调用该子程序时就以该行号为依据。

子程序必须以返回指令“RETURN”语句结束。“RETURN”在英文中为返回的意思，它的作用是控制子程序返回原调用程序的下一条语句继续执行程序。

2. **子程序的调用**　调用子程序的指令为“GOSUB”，它是“转去执行子程序”的英文缩写(GO to SUBroutine)。“GOSUB”后面必须跟着被调用子程序的入口行号，如“GOSUB 100”等。

3. **程序No. 3－8的优化**

(1)归纳子程序

分析程序 No. 3－8，其排序和输出部分的程序 30～70 行、140～180 行、240～280 行三段程序基本相同，可归纳成子程序，见程序 No. 3－9。

No. 3－9

```
500 REM Pai xu shu chu SUB
510 FOR K=1 TO 9: FOR J=K+1 TO 10
520 IF D(J)>D(K) THEN SWAP A(J), A(K): SWAP D(J), D
    (K)
530 NEXT: NEXT
540 CLS:PRINT A$ : PRINT
550 FOR I=1 TO 10: PRINT "DI";I;"MING";A(I); D(I):
    NEXT
560 RESTORE: A$=INKEY$(0)
570 RETURN
```

子程序中，我们把排序种类提示语句行中的字符串改为字符串变量 A$，这样在调用子程序之前，只要对串变量 A$ 赋以不同的值就可以改变排序种类的提示内容。

(2)优化主程序

依据子程序 No. 3－9，可把多科排序程序 No. 3－8 优化为 No. 3－10 的形式(并连接子程序 No. 3－9)。

```
5 REM No. 3－10
10 CLS: DIM A(10),D(10)
20 FOR I=1 TO 10: READ A,B,C: A(I)=A: D(I)=B+C:
   NEXT
30 A$="ZONG CHENG JI PAI XU": GOSUB 500
40 FOR I=1 TO 10: READ A,B,C: A(I)=A: D(I)=B: NEXT
50 A$="SHU XUE CHENG JI PAI XU":GOSUB 500
60 FOR I=1 TO 10: READ A,B,C: A(I)=A: D(I)=C: NEXT
70 A$="YU WEN CHENG JI PAI XU": GOSUB 500
```

```
80 END
90 DATA 1000,89,83,1001,97,82,1002,67,93,1003,84,87,
   1004,99,86,1005,71,96,1006,94,78,1007,86,89,1008,77,
   96,1009,100,98
500 REM Pai xu shu chu SUB
510 FOR K=1 TO 9: FOR J=K+1 TO 10
520 IF D(J)>D(K) THEN SWAP A(J),A(K): SWAP D(J),D
   (K)
530 NEXT: NEXT
540 CLS: PRINT A$: PRINT
550 FOR I=1 TO 10: PRINT "DI";I;"MING"; A(I); D(I):
   NEXT
560 RESTORE: A$=INKEY$(0)
570 RETURN
```

主程序中 20、30 行程序完成总成绩的排序和输出；40、50 行程序完成数学成绩的排序和输出；60、70 行程序完成语文成绩的排序和输出。

在有子程序结构的程序中，程序尾的“END”语句行极为重要。若没有它，程序将执行完主程序后又执行到子程序中，到最后遇到“RETURN”语句时，因无“GOSUB”语句调用而运行出错。读者可试删除 80 行，观察所造成的后果。

现在运行程序 No. 3－10，将打印出与程序 No. 3－8 相同的结果。但两个程序相比，No. 3－10 要简洁多了。

第七节　本章小结

本章共讨论了 INKEY $、INPUT、CLS、NEW、DIM、SWAP、READ、DATA、RESTORE、REM、GOSUB、RE-

TURN 十二条指令，现将各条指令综述如下：

1. 键盘扫描指令(INKEY $)

格式：INKEY $[(n)]

类型：函数

简写：INK.

功用：对键盘进行扫描，等待读取输入的某一字符。如果在该函数的执行期间，用户按下了某一个按键，则该键的字符就会赋给 INKEY $ 函数。这样，程序就可以根据 INKEY $ 取值情况，决定转移的方向或进行哪一项操作。

注释：INKEY $ 取自于“INputfromKEYboard”，是“从键盘上输入”的意思。

INKEY $ 与 INPUT 指令不同，它无须等待用户按回车键就可以读取一个字符。

说明：

(1)n 是一个亚变量，它可以取 −32768～32767 之间的任意数值，也可以是变量或表达式(采用变量或表达式时无实用意义)。

(2)n 值可以省略。当省略 n 值时，若用户不按键则函数值为零，并继续执行 INKEY $ 以后的语句；当使用 n 时，则程序在执行到 INKEY $ 语句时，暂停运行，等待用户从键盘上输入某个字符后才向下执行。

示例：

```
10 REM INKEY $
20 A $ =INKEY $
30 IF A $ <>" "THEN PRINT A $ : GOTO 20
40 PRINT "NO";: GOTO 20
```

运行程序，将连续横向打印“NO”，按哪一个按键时，将纵

向打印该按键的符号。示例程序说明，INKEY $ 函数省略 n 后，程序执行到 INKEY $ 语句的时刻没有按键时，向下执行 40 行，打印“NO”；有按键时则执行 30 行后面的语句，打印按键的符号。

2. **键盘输入指令(INPUT)**

格式：INPUT“提示内容”；变量 1，变量 2，变量 3，……

类型：语句或命令

简写：I.

功用：接收来自于键盘输入的数据并把它们赋给相应的变量。INPUT 指令主要适用于变量取值事先无法确定或同一程序处理不同数据的场合。

注释：INPUT 在英文中为“输入”的意思。

语句中提示内容是由用户设定的提示字符串，这些字符串必须使用双引号括起。当程序执行到 INPUT 语句时，屏幕上就会显示出这些字符的信息，提示用户输入相应的数据。若在输入语句时省略了提示内容和后面的分号，当执行到该语句时只显示一个问号，无其它提示信息。

当提示内容后不用分号而改用逗号时，在执行 INPUT 语句时将不显示问号。

变量就是 INPUT 语句所要赋值的对象，数值或字符串均可。一个 INPUT 语句可以含两个以上的变量。当输入多个变量时，变量之间要用逗号隔开。

说明：使用 INPUT 语句输入的数据既可以是数值型也可以是串型，但输入的变量类型要一一对应。若输入了类型不同的变量电脑会做出不同的处理。如在要求输入字符串时而输入了数字，则把数字当做字符串处理；若在要求输入数值时而输入了字符串，则把字符串当做 0 处理。

3. **清屏指令(CLS)**

格式:CLS

类型:语句或命令

简写:CL.

功用:清屏,并使光标定位于屏幕左上角。

清屏之后,用户就可以在屏幕上显示新的信息。

注释:CLS 源于"CLear Screen",是"清除屏幕"的意思。

说明:

(1)CLS 语句仅限于清屏,并不会清除程序中的变量。

(2)CLS 一般编写在程序中作为语句使用。作为命令使用的场合不太多(因即时清屏一般使用清屏操作命令较方便)。

4. **内存清除命令(NEW)**

格式:NEW

类型:命令

功用:清除内存中的当前程序,将全部数值变量置 0,将串变量置空。这一命令通常在输入新程序前,为了避免现有的程序对新程序干扰的情况下使用。

注释:NEW 在英文中为"新"的意思。在此表示把系统置为新的状态(即未使用的状态),也就是清除内存。

5. **数组说明语句(DIM)**

格式:DIM<数组名 1(元素数目)>,<数组名 2(元素数目)>……

类型:语句

简写:DI.

功用:建立或定义数组,即在内存中为数组开辟存储空间,确定数组名、数组数目、数组类型、数组维数及各维元素数

目。

注释:DIM 源于“DIMension”,“维数”的意思。

数组名代表所要定义的数组,它与变量名类似。但数组与普通的数值变量或字符串又有区别,它里面存放的是若干个有序元素,每个元素都是一个变量。这些元素或者是数值型的,或者均是字符型。如果是字符型的,定义数组名时要像定义串变量那样带上“$”符号。DIM 语句一次可以只定义一个数组,也可以定义多个数组。

在对每个数组进行定义时,也要同时定义数组的维数。维数的多少,是由 DIM 语句中各数组名之后所跟的元素数目项的多少决定的,如 A(13),元素数目项数为 1,所以它是一维数组;A(2,3)则是二维数组。同时 A(13)也表示一维数组所能容纳的元素个数是 14 个(最小下标变量是 A(0)、最大下标变量是 A(13))。可见,对于一维数组 A(m),其变量个数为 m+1 个。同理,对于二维数组 A(m,n),变量个数为(m+1)*(n+1)个。

说明:

(1)数组必须先定义后使用,因此 DIM 语句一般安排在程序的前面。

(2)在同一程序中,同一名称数组只能定义一次,除非使用 CLEAR 语句清除,否则不能重复定义。

(3)数组的每维元素数目限定在 255 个以内。

6. 变量内容交换指令(SWAP)

格式:SWAP<变量 1>,<变量 2>

类型:语句

简写:SW.

功用:两个变量之间相互交换内容。

利用这一语句可以对两个或两个以上的变量进行比较或对多个数据进行排序。

注释:SWAP 为“交换或者对调”的意思。

变量 1 和变量 2 是其值将要互相交换的两个变量。该语句执行后,前者的原值便会送给后者,后者的原值便会送给前者。

变量 1 和变量 2 在类型上要保持一致,或者均为串型,或者均为数值型。

7. **读数、置数指令(READ、DATA)**

格式:READ<变量 1>,<变量 2>,……

DATA<常量 1>,<常量 2>,……

类型:语句

简写:REA.

D.

功用:READ 语句从 DATA 语句中读取数据,并把它赋给相应的变量。

DATA 语句为 READ 语句中的变量提供数据。

READ、DATA 语句适用于对较多变量赋值的场合。DATA 语句的特点是一次可以提供一批数据。

注释:READ 是“读”的意思,DATA 是“数据”的意思。

说明:

(1)DATA 是非执行语句,它可以放在程序中的任意位置,但若有多个 DATA 语句时,它们之间必须有先后顺序。另外 DATA 语句提供的数据个数不能少于 READ 中变量的个数。

(2)DATA 语句中的数据可以是数值、字符串,但不能是变量、函数或表达式。DATA 语句中的数据类型必须与

READ 语句中的变量类型一一对应。

8. 恢复数据区指令(RESTORE)

格式:RESTORE<行号>

类型:语句

简写:RES.

功用:将数据区的指针恢复到某个 DATA 语句上。

READ 语句的变量每读取一项数据,DATA 数据表中的数据指针便向下移动一个数据项位置。当 DATA 数据表中的数据均被读完之后,指针便指到数据区的末尾。如果再次读取数据,指针就需要恢复到前面某个数据项的位置上,否则将出错。RESTORE 语句为我们反复使用同一批数据资源提供了方便。

注释:RESTORE 为"恢复数据区"的意思。

语句中的行号代表某个 DATA 语句。如果省略行号则数据指针恢复到整个数据区首项数据位置上,下一个 READ 语句将从程序的第一个 DATA 语句的第一项数据开始重读数据。如果选用行号则数据指针便恢复到该行号所指定的 DATA 语句行的首项数据位置。

9. 程序注释指令(REM)

格式:REM<注释内容>

类型:语句

简写:,

功用:对语句或程序段的作用作简要说明,以增强程序的易读性。

注释:REM 源于"REMark",是"注释、注解或说明"的意思。

注释内容可以是任何文字、字符,甚至可以是 F BASIC

的保留字。注释内容仅仅是给编程序或阅读程序的人看的，程序运行后，F BASIC 对注释语句的内容不予理会，故 REM 为非执行语句。

最长注释内容以 255 个字符为限。

说明：

(1)在多重语句行(一个行号下含多条语句)中，只有 REM 之前的语句才能执行，而同行中 REM 后的语句全作为注释内容对待，不予执行。

(2)如对多个语句进行注释，则每个被注释的语句标号都要写上 REM。

(3)注释语句虽然为非执行语句，但在用 LIST 指令显示程序清单时，仍会显示该语句的全部内容。

(4)GOTO、GOSUB 等语句可以转移到 REM 语句上，但从 REM 后第一个可执行的语句处开始向下执行。

10. **子程序调用、返回指令(GOSUB、RETURN)**

格式：GOSUB＜行号＞

RETURN [行号]

类型：语句

简写：GOS.

RE.

功用：GOSUB 将程序的控制权交给指定的子程序；

RETURN 将程序的控制权交还给调用程序。

子程序可以完成程序的某一部分功能。

采用转子的方法可使程序的结构和层次清晰，便于阅读和修改。

注释：GOSUB 取自于“GO to SUBroutine”，是“转去执行子程序”的意思。

RETURN 为“返回”的意思。

行号为转移或返回的目标，它代表指定的程序。行号要取数值常数形式，取值范围为 0～65534。

子程序的最后一条语句必须是返回语句 RETURN。RETURN 后的行号为可选项，用于子程序返回的位置。如果省略行号则返回到调用程序中 GOSUB 语句的下一语句。

RETURN 必须与 GOSUB 语句联用，否则，当无 GOSUB 语句而执行到 RETURN 语句时将中断程序运行，显示出错信息。

第四章
飞奔的玛丽翱翔的鹰

F BASIC 语言最突出的特点就是方便灵活地控制卡通运动。本章将重点介绍与卡通的定义和运动控制有关的十三条指令：

打开卡通显示页开关指令	SPRITE ON
关闭卡通显示页开关指令	SPRITE OFF
定义显示卡通指令	DEF SPRITE
显示卡通显示定位指令	SPRITE n,X,Y
显示卡通消失控制指令	SPRITE n
定义运动卡通指令	DEF MOVE
运动卡通运动坐标设定指令	POSITION
运动卡通运动启动指令	MOVE
运动卡通运动状态测试指令	MOVE(n)
控制运动卡通停止指令	CUT
控制运动卡通消失指令	ERA
运动卡通坐标测试指令	XPOS
	YPOS

并根据编程的需要附带介绍：

操纵器方向钮测试指令	STICK
操纵器功能钮测试指令	STRIG

光标定位指令	LOCATE
延时指令	PAUSE
ASCII 码——字符转换指令	CHR $

第一节　迷人的卡通图案

卡通——就是电脑游戏中可动的动画，如乌龟、螃蟹、飞鸟等动物，汽车、飞机、火球等物体，还有勇士、小姐等人物。本书封四展示的就是 F BASIC 语言系统中已固化的卡通图案，这些图案的名称依次是：玛丽、丽莎、苍蝇、飞鱼、妖怪、企鹅、火球、汽车、太空站、杀手卫星、太空船、爆炸、乌龟、螃蟹、飞鸟、激光共 16 类卡通（激光卡通又分激光弹、激光束、激光刀）。仔细观察卡通图案，在每个卡通的四角上分别有 4 个号码，这些号码称为卡通图案代码，它们与附录一中的代码完全对应。当我们需要某个卡通图案时，只要在特定的指令中写入其代码，就可以显示对应的卡通图案。

由封四可见，同一个卡通有多种姿势。如玛丽、丽莎有七种姿势，飞鱼、汽车有六种姿势，乌龟、螃蟹、飞鸟等只有两种姿势。这些姿势的设计就是为了逼真地模仿出卡通的动画效果，如玛丽走必须由走 1、走 2、走 3 三种姿势交替出现才能完成“走”的动作，而乌龟走只须立正、走两种姿势就可完成。

卡通、卡通动作和卡通姿势是三个重要的概念，它们的从属关系是：

卡通姿势从属于卡通动作，卡通动作从属于卡通。也就是说由卡通姿势组成卡通动作，由卡通动作构成生灵活现的卡通。

第二节　独特的显示方式

独特的显示方式是F BASIC语言的又一特点。它的显示画面由四层画面叠加而成，如图4—1所示。

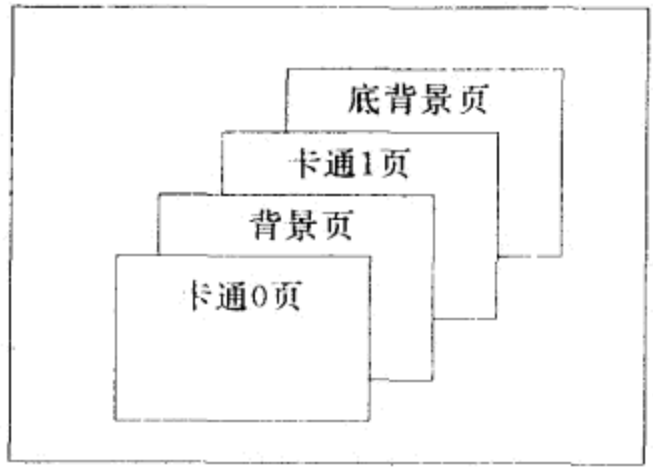


图4—1　F BASIC显示画面的结构

1. **卡通0页**　第一层称为卡通0页。显示在该页面上的卡通可以遮住背景页的对应部分的内容，形成的效果是卡通在背景前面显示。卡通页面的尺寸是按点计算的，其大小为：

X方向(横向)从左向右为0～255点

Y方向(纵向)从上到下为0～239点

即卡通页的尺寸为256×240点。

2. **背景页**　第二层称为背景页。F BASIC的指令和程序都在这一页面上显示。背景图形(见封二)也将显示在这一页面上。背景页的尺寸按字符计算(每个字符为8×8点)，其大小为：

X 方向(横向)从左向右为 0～27 列

Y 方向(纵向)从上到下为 0～23 行

即背景页的尺寸为 28 列×24 行。

注——在任天堂游戏中背景页的实际尺寸为 32 列×30 行，在 F BASIC 中，考虑到电视屏幕边缘不利于观察显示的程序，左右各压缩两列变为 28 列，上下各压缩 3 行变为 24 行。

3. **卡通 1 页** 第三层称为卡通 1 页，在某些特殊情况下，卡通的部分或全部需要隐藏在背景之后，这时就要控制卡通显示在卡通 1 页上面。这一页的大小与卡通 0 页相同。卡通 0 页、卡通 1 页统称为卡通页。

4. **底背景页** 第四层称为底背景页。这一页面通常用于显示蓝天、大海、森林、沙漠等大面积、单一色调的效果。该页只能显示一种颜色，F BASIC 的初始画面被设定为黑色，可使用指令把底背景设定为任意颜色(第五章中介绍)。底背景页的大小为：

X 方向 32 个字符长

Y 方向 30 个字符宽

即其显示尺寸为与卡通页相同。

5. **显示页面间的对应关系** 因为背景以字符为显示单位，卡通以点为显示单位，两个页面的量纲不统一，又由于两个页面的大小不同，致使两个页面间存在着换算关系。在游戏程序的编写中，必须了解卡通页和背景页的对应关系，才能正确做出卡通运动和游戏的进程判断。

卡通、背景显示页面间的对应关系见图 4－2。

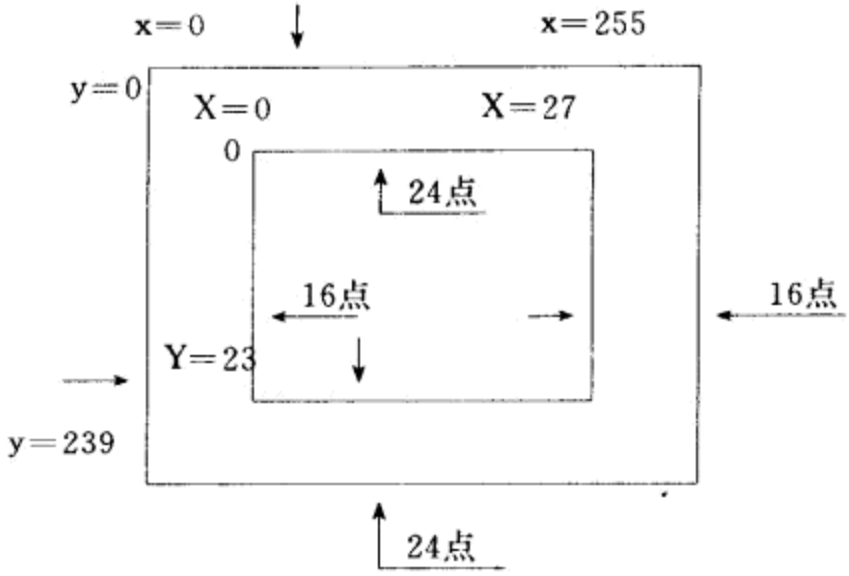


图 4－2　卡通、背景页对应关系图

根据图 4－2，若定义“x、y”代表卡通页的坐标，“X、Y”代表背景页的坐标，可写出两个页面间的换算公式。

已知背景页坐标换算卡通页的坐标公式：

$x=(X*8)+16$

$y=(Y*8)+24$

已知卡通页坐标换算背景页的坐标公式：

$X=(x-16)/8$

$Y=(y-24)/8$

第三节　潇洒的玛丽兄弟

一、卡通显示页的开关

卡通显示页必须打开后才能显示卡通，开显示指令为“SPRITE ON”，其中“SPRINT”在英文中为“幽灵、小精灵”

的意思，F BASIC 取来表示卡通。“SPRITE ON”就是令卡通显示的意思。

相应地卡通显示页也可以关闭，关闭卡通页的指令为“SPRITE OFF”。

二、玛丽出场

请对照输入程序 No. 4－1。

```
10 REM No. 4－1
20 CLS: SPRINT ON
30 DEF SPRITE 0,(0,1,0,0,0)=CHR$(0)+CHR$(1)+CHR
   $(2)+CHR$(3)
40 SPRITE 0,100,100
```

运行程序，将在画面上显示封底玛丽走 1 的图象。

若想显示玛丽立正的图象，只须把 30 行中四个“CHR $()”的数字依次换为“4、5、6、7”即可(参见封底玛丽走 2)。

看来，叫玛丽出场极其容易。

程序 No. 4－1 中使用了三条新指令，DEFSPRITE、CHR $、和 SPRITE。

1. **定义显示卡通** DEF SPRITE 是一个组合指令，它的作用是定义显示卡通。在 DEF SPRITE 构成的语句中，等号以前的部分定义显示卡通的状态，等号以后的部分定义显示卡通的类别。该语句的通用形式如下：

DEF SPRITE n,(A,B,C,D,E)=<字符表达式>

语句中“n、A、B、C、D、E”六个参数的含义是：

n——显示卡通的序号。F BASIC 允许使用 DEFSPRITE 语句最多定义 8 个显示卡通，序号为 0～7。在程序 No. 4－1 中，我们定义了 0# 显示卡通。

A——显示卡通的配色号码(或称代码),取值范围0～3。No.4－1中选用了0#配色代码,显示的玛丽为蓝色。若改为1、2、3#配色时,显示的颜色将依次为黄色、红色和绿色,当学习了配色指令后,可以令玛丽显示任意颜色。有关配色号码的设定将在下一章讨论。

B——卡通的大小,取值为0、1。B=0时显示的卡通为8×8点阵,相当于一个字符哪样大;B=1时显示卡通为16×16点阵,相当于四个字符大。No.4－1中定义的玛丽是16×16点阵的,若改为0,将仅显示玛丽的半个头部。

C——选择卡通显示的页面(也称卡通显示优先度),取值为0、1。取值为0时将在卡通0页上显示;取值为1时将在卡通1页上显示。当没有背景图案时,在0、1页显示的卡通没有区别。

D——卡通在X轴方向的翻转指示,取值为0、1。取值为0时定义的卡通面向与封底的图形一致(面向左);当取值为1时,卡通的面向将与封底的图形相反。

注意——在定义横向翻转的卡通时,必须将等号后的卡通序号左右颠倒填入。如把No.4－1中的30行改为

```
30 DEF SPRITE 0,(0,1,0,1,0)=CHR$(1)+CHR
    $(0)+CHR$(3)+CHR$(2)
```

将显示面向右的玛丽。

E——卡通在Y轴方向的翻转指示,取值为0、1。取值为0时卡通在垂直方向的显示与封底图形一致;取值为1时,垂直方向的显示将与封底相反。

注意在定义纵向翻转的卡通时,必须把等号后的卡通序号上下颠倒填入。如把No.4－1中的30行改为

```
30 DEF SPRITE 0,(0,1,0,0,1)=CHR$(2)+CHR
```

$(3)+CHR$(0)+CHR$(1)

将显示面向左、头向下的玛丽。

如把 No. 4－1 中的 30 行改为

```
30 DEF SPRITE 0,(0,1,0,1,1)=CHR$(3)+CHR$(2)+CHR
   $(1)+CHR$(0)
```

将显示面向右、头向下的玛丽。

以上六个参数，可以取常数、变量或表达式。

2. **把 ASCII 码转换为字符** 附录三所列的符号代码表称为 ASCII 码表，“ASCII”为“美国国家标准信息交换代码”(American Standard Code for Information Interchange)的英文缩写，它规定了计算机语言中每个字符所对应的代码，如大写英文字母“A”的代码为 65，小写英文字母“a”的代码为 97 等。一般把“A”等字符称为 ASCII 字符，它是计算机对字符的外部显示形式；把每个字符所对应的代码称为 ASCII 码，它则是计算机内部对字符的表示形式。如组成玛丽走 1 的四个卡通图形代码是“0、1、2、3”，它们是电脑内部的表示形式。当我们要显示玛丽走 1 时，则必须把“0、1、2、3”所对应的四个图形进行还原。还原操作就由 ASCII 码转换为字符指令“CHR$”来完成。

“CHR$”指令的语句形式为

CHR$(X)

语句中的“X”就是 ASCII 代码，取值范围为 0～255 之间的整数，它也可以是变量或表达式。我们若把程序 No. 4－1 中四个“CHR$”指令中的参数换为其它数据，将显示其它卡通。

由于符号、代码一一对应，在显示卡通定义语句等号后的字符表达式也可以用代码对应的字符表示。如“80、81、82、83”为“飞鱼前 1”姿势的四个图形代码，而从附录三可见，它

们又是英文大写字母“PQRS”的代码，故程序 No. 4－1 的 30 行又可以写为：

```
30 DEF SPRITE 0,(0,1,0,0,0)="PQRS"
```

运行程序将显示向前飞的飞鱼卡通。

3. **显示卡通的定位** 由 DEF SPRITE 语句定义了显示卡通的状态和类型后，剩下的问题就是决定卡通的显示位置（即显示坐标）。“SPRITEn,X,Y”就是定义显示卡通坐标的专用语句，其三个参数的含义是：

n——显示卡通的序号，它与 DEFSPRITE 语句中的卡通序号相对应。

X——显示卡通的 X 坐标，取值范围 0～255。

Y——显示卡通的 Y 坐标，取值范围 0～239。

当 SPRITE 语句省略了后面的两个坐标参数时，其作用是令“n”号显示卡通消失。

三、玛丽兄弟手拉手

有以上三条语句垫底，我们就可发挥自己的想象力，定义显示出手拉手的玛丽兄弟，程序见 No. 4－2。

```
10 REM No. 4－2
20 CLS: SPRINT ON
30 DEF SPRITE 0,(1,1,0,0,0)=CHR$(8)+CHR$(9)+CHR
   $(10)+CHR$(11)
40 DEF SPRITE 1,(2,1,0,1,0)=CHR$(9)+CHR$(8)+CHR
   $(11)+CHR$(10)
50 SPRITE 0,100,100
60 SPRITE 1,86,100
```

运行程序，将显示红衣服哥哥领着黄衣服弟弟的画面。

第四节　让玛丽哥哥扭起来

一、让玛丽哥哥扭一圈

上一节唤出了玛丽兄弟，本节讨论让玛丽哥哥扭起来的编程方法。

由封底看到，玛丽走有三种姿势，“走 1”是玛丽一脚迈出的姿势、“走 2”是立正的姿势、“走 3”是前脚落地后脚抬起的姿势，若令这三种姿势交替出现，并且每一种姿势出现时在 X 方向上有一定的位移，则可模仿出玛丽走路的模样。根据这一思路，编写的程序见 No. 4－3。

```
10 REM No. 4－3
20 CLS: SPRITE ON
30 FOR I=0 TO 2
40 DEF SPRITE I,(0,1,0,1,0)=CHR$(I*4+1)+CHR$(I*
   4)+CHR$(I*4+3)+CHR$(I*4+2)
50 NEXT
60 X=10: Y=100
70 X=X+2: SPRITE 0,X,Y: SPRITE 2: PAUSE 10
80 X=X+2: SPRITE 1,X,Y: SPRITE 0: PAUSE 10
90 X=X+2: SPRITE 2,X,Y: SPRITE 1: PAUSE 10
100 IF X<240 THEN 70
```

程序说明：

1. 程序中 70～90 行的“PAUSE”语句是一条新指令，它在作用是控制程序暂停一段时间后再继续执行，延时的长短由“PAUSE”指令后的参数决定，延时参数的取值范围为 0～32767。

2. 程序中的 30～50 行定义了玛丽走 1、走 2、走 3 三种姿势的显示卡通。

3. 60～100 行控制玛丽交替显示。显示原理是：

(1)令 0# 卡通(玛丽走 1)显示，同时令 2# 卡通消失；

(2)0# 卡通延时显示 10 个单位后，1# 卡通(玛丽走 2)显示，0# 卡通消失；

(3)1# 卡通显示延时 10 个单位后，2# 卡通显示，同时 1# 卡通消失；

(4)2# 卡通显示 10 个单位后，转(1)重复执行以上过程。直到卡通的 X 坐标等于 240 时结束。

运行程序，只见玛丽哥哥由屏幕左侧扭向右侧而停止。

二、玛丽哥哥绕圈扭

程序 No. 4－3 靠改变玛丽三个姿势在 X 方向的显示坐标，实现了玛丽横向行走的控制，若同时改变 Y 坐标则可以控制玛丽向任意方向行走。如程序 No. 4－4 可控制玛丽由屏幕的左上方开始向右下方周尔复始的行走。

```
10 REM No. 4－4
20 CLS: SPRITE ON
30 FOR I=0 TO 2
40 DEF SPRITE I,(0,1,0,1,0)=CHR$(I*4+1)+CHR$(I*
   4)+CHR$(I*4+3)+CHR$(I*4+2)
50 NEXT
60 X=10: Y=0
70 X=X+2: SPRITE 0,X,Y: SPRITE 2: PAUSE 10
80 X=X+2: SPRITE 1,X,Y: SPRITE 0: PAUSE 10
90 X=X+2: SPRITE 2,X,Y: SPRITE 1: PAUSE 10
100 IF X<240 THEN 70
```

```
110 GOTO 60
```

在程序 No. 4－3、No. 4－4 中，改变卡通坐标 X 或 Y 的增量和调整延时语句的参数，可灵活地控制卡通运动的速度。

三、让玛丽扭的更潇洒

1. 运动卡通的概念 以上使用了“DEF SPRITE”语句定义了显示卡通，并通过改变其显示位置和姿势的方法实现了卡通运动控制。这种卡通运动控制处理程序中，既要考虑卡通的显示姿势、显示坐标，还要安排每种姿势的显示时间和消失时机，程序编写较繁琐。下面要讨论的是 F BASIC 中的另一类卡通——运动卡通。

运动卡通的运动控制较简单，我们只要在程序中说明运动卡通的运动方向、运动速度、运动距离和运动前的坐标等几个参数就可以了，剩下的工作由系统控制完成。

2. 运动卡通的示例程序 我们仍采用先接触程序再介绍指令的方式讨论运动卡通的定义过程。程序见 No. 4－5。

```
10 No. 4--5
20 CLS: SPRITE ON
30 DEF MOVE(0)=SPRITE(0,3,1,255,0,0)
40 POSITION 0,127,120
50 MOVE 0
```

运行程序，蓝色的玛丽哥哥在屏幕中间开始以飞快的速度向右奔跑，跑出屏幕的右边界后，又从屏幕的左边界进入，奔跑两圈后在出发点处停止。

3. 程序说明 程序中 30 行为 0# 运动卡通运动状态的说明语句，40 行为 0# 运动卡通的运动起始位置说明语句，50 行为 0# 运动卡通运动的启动语句。

4. **运动卡通的定义说明** DEF MOVE 为运动卡通的定义指令，由该指令组成的语句说明了运动卡通的编号、运动卡通的种类、运动方向、运动速度、运动量、显示优先度和卡通的显示配色。该指令的语句通用形式为

DEF MOVE(n)=SPRITE(A,B,C,D,E,F)

语句中共有七个参数，含义如下：

n——定义运动卡通的序号(注意它与显示卡通的序号是两个概念)。F BASIC 允许一次定义 8 个运动卡通，编号为 0～7。

A——运动卡通种类的编号，取值范围为 0～15。编号与卡通种类的对应关系见表 4－1。

表 4－1 运动卡通种类与编号一览表

编 号	卡通种类	编 号	卡通种类
0	玛 丽	8	杀手卫星
1	丽 莎	9	太空船
2	苍 蝇	10	爆 炸
3	飞 鱼	11	妖 怪
4	企 鹅	12	激 光
5	火 球	13	乌 龟
6	汽 车	14	螃 蟹
7	太空堡垒	15	飞 鸟

B——运动卡通的运动方向代码，取值 0～8。运动方向与

代码的对应关系见表 4－2、图 4－3。

表 4－2　运动卡通的运动方向代码一览表

代　码	运动方向	代　码	运动方向
0	静止不动	5	下
1	上	6	左下
2	右上	7	左
3	右	8	左上
4	右下		

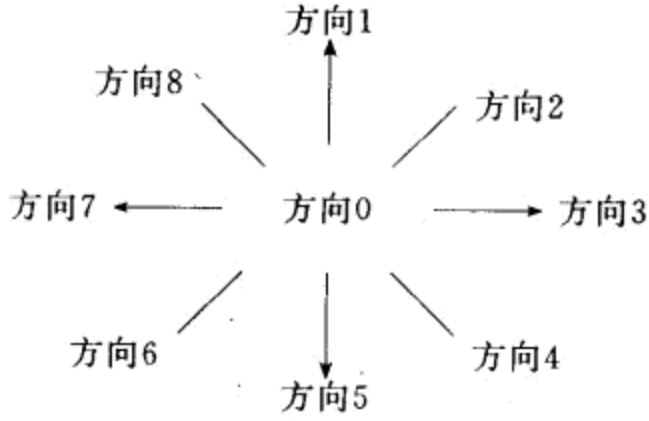


图 4－3　运动卡通的运动方向示意图

C——运动卡通的运动速度，取值范围为 1～255。等于 1 时速度最快，等于 255 时速度最慢。C 的取值与速度的对应关系为：

$$V=60/C \quad (点/秒)$$

速度 V 的量纲为“点/秒”，即每秒钟移动“V”点。如

C＝1　V＝60，即 1 秒钟移动 60 点

C＝3　V＝20，即 1 秒钟移动 20 点

C=60　V=1，即1秒钟移动1点

D——运动卡通的一次移动量，取值范围为1～255，一次移动量是指定义的运动状态一次移动的距离。

卡通在平行、垂直方向上运动时，D的取值与移动距离(S)的对应关系为：

S=2×D（点）

如

D=1　移动2点

D=100　移动200点

程序No.4－5中D=255，故玛丽移动510点。

当卡通在斜方向上运动时，移动距离与D的对应关系为：

S=2.828×D（点）（运算后取整）

注意——D不能等于0，否则卡通不能显示。

E——运动卡通显示优先度，取值0、1。等于0时从背景前面显示；等于1时从背景后面显示。

F——运动卡通的配色号码，取值为0～3。

以上七个参数均可以取常量、变量和表达式。

5. **运动卡通的运动起始坐标说明**　"POSITION"为运动卡通的运动起始坐标说明指令，其语句格式为：

POSITION 0n,X,Y

三个参数的含义如下：

n——DEF MOVE语句中定义的运动卡通序号，取值为0～7。

X——卡通运动起点的横坐标，取值范围0～255。

Y——卡通运动起点的纵坐标，取值范围0～255。

6. **卡通运动的启动** “MOVE”为卡通运动的启动指令，语句格式为

MOVE n

参数“n”为 DEF MOVE 语句中定义的运动卡通序号，取值为 0～7。

四、让玛丽飞奔不休

程序 No. 4－5 只能控制玛丽运动两圈而停止，怎样令玛丽奔跑不停呢？请看程序 No. 4－6

```
10 No. 4－6
20 CLS: SPRITE ON
30 DEF MOVE(0)=SPRITE(0,3,1,255,0,0)
40 POSITION 0,127,120
50 MOVE 0
60 IFMOVE(0)=－1 THEN 60
70 GOTO 50
```

运行程序

玛丽飞奔，周尔复始……。

分析程序 No. 4－6，它只比 No. 4－5 多出了 60、70 两行，其中 60 行的“MOVE(0)”起到了关键的作用。

“MOVE(0)”是运动卡通运动是否完成的测试语句，当运动卡通在运动中时

MOVE(0)＝－1

程序在 60 行反复执行；当运动卡通的运动完成时

MOVE(0)＝0

这时执行 70 行，转到 50 行再启动 0＃运动卡通运动。如此周尔复始，控制玛丽连续运动。

五、玛丽八兄弟

八个玛丽排成一列。

是正步演习？

是竞走比赛？

还是马拉松……？

请执行程序 No. 4－7。

```
10 No. 4－7
20 CLS: SPRITE ON
30 FOR I=0 TO 7
40 DEF MOVE(I)=SPRITE(0,3,1,255,0,I/2)
50 POSITION I,127,I * 25+20
60 NEXT
70 FOR I=0 TO 7: MOVE I: NEXT
80 IF MOVE(0)=－1 THEN 80
90 GOTO 70
```

程序 No. 4－7 仅是把定义一个运动卡通的程序 No. 4－6 扩展到八个卡通，没啥出奇。但它看似平凡，却内蕴多种变化。如改 40 行为

40 DEF MOVE(I)=SPRITE(1,3,1,255,0,I/2)

可调来丽莎八姊妹。当然也可以调动八个苍蝇、八只飞鸟、八只企鹅、八只乌龟……。

若把 40 行改为

40 DEF MOVE(I)=SPRITE(I,3,1,255,0,I/2)

将有玛丽、丽莎、苍蝇、飞鱼、妖怪、企鹅、火球、汽车八类卡通列队竞赛，当然也可以调出另外的八类卡通。

若把 40 行改为

40 DEF MOVE(I)=SPRITE(I,I+1,I+1,255,0,I/2)

将有八类卡通，分别以八种速度、向八个方向运动。它们有的飞奔、有的蹒跚、有的急急忙忙、有的从容不迫、有的闲庭信步、有的扭扭捏捏……，你不感到奇妙吗？

第五节　玛丽哥哥走八角

一、问题

图 4－4 是一个八边形，试编写程序令玛丽沿八边形走一圈。

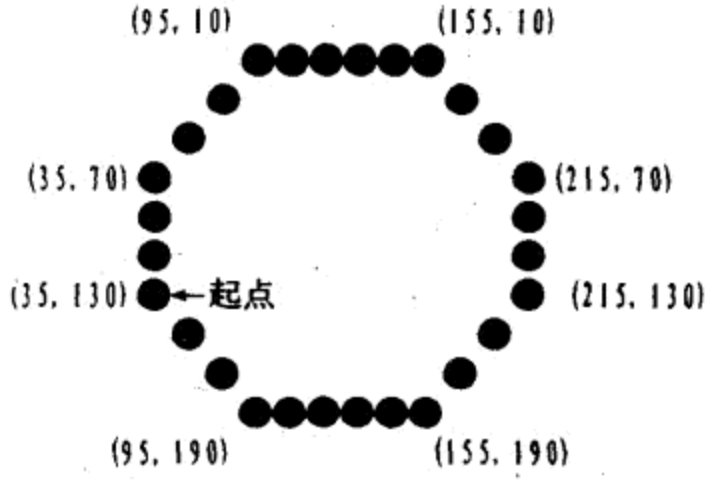


图 4－4　八边形

二、编写程序

由图 4－4 可见，八边形两角间的距离是 60 点，因此可以确定玛丽的运动距离参数是 30。运动距离确定后，只要按以上定义运动玛丽的编程方法，控制玛丽从各个角开始沿边的伸展方向运动 30 点，八个方向全部走完，玛丽就回到了起点。

程序见 No. 4－8。

```
10 REM No. 4－8
20 CLS: SPRITE ON
30 DEF MOVE(0)=SPRITE(0,1,1,30):
   POSITION 0,35,130:MOVE 0
40 DEF MOVE(0)=SPRITE(0,2,1,30):
   POSITION 0,35,70:MOVE 0
50 DEF MOVE(0)=SPRITE(0,3,1,30):
   POSITION 0,95,10:MOVE 0
60 DEF MOVE(0)=SPRITE(0,4,1,30):
   POSITION 0,155,10:MOVE 0
70 DEF MOVE(0)=SPRITE(0,5,1,30):
   POSITION 0,215,70:MOVE 0
80 DEF MOVE(0)=SPRITE(0,6,1,30):
   POSITION 0,215,130:MOVE 0
90 DEF MOVE(0)=SPRITE(0,7,1,30):
   POSITION 0,155,190:MOVE 0
100 DEF MOVE(0)=SPRITE(0,8,1,30):
    POSITION 0,95,190:MOVE 0
```

运行程序，将看到玛丽哥哥从起点开始，匆匆忙忙地走完了八个边，又回到起点待命。

三、程序优化

上述程序重复语句太多，有没有更好的编程方法呢？

请看程序 No. 4－9。

```
10 No. 4－9
20 CLS:SPRITE ON
30 LOCATE 9,9: PRINT "Ma Li ge ge"
40 LOCATE 6,10:PRINT "Zou ba bian xing"
```

```
50 X=35: Y=130
60 FOR I=0 TO 7
70 DEF MOVE(0)=SPRITE(0,I+1,1,30)
80 POSITION 0,X,Y:MOVE 0
90 IF MOVE(0)=-1 THEN 90
100 X=XPOS(0): Y=YPOS(0)
110 NEXT
```

运行程序，玛丽哥哥除做出了与程序 No. 4－8 相同的运动之外，还在屏幕中间出现了“玛丽哥哥走八边形”的汉语拼音字幕。

四、优化程序分析

程序 No. 4－9 简练、有效，但不好理解。下面讨论编程原理。

(一)新指令介绍

程序中出现了三个新面孔“LOCATE、XPOS、YPOS”，它们起什么作用呢？

1. **字符显示定位** “LOCATE”指令是字符显示定位指令，它是为“PRINT”指令服务的指令，当要在屏幕上的某一位置显示某些内容时，就要使用 LOCATE 指定显示位置。LOCATE 指令的语句格式为：

LOCATE X,Y

语句中的两个参数就是显示位置的横、纵坐标，X 的取值范围为 0～27，Y 的取值范围为 0～23。

2. **运动卡通当前坐标的测试** 通过“XPOS、YPOS”指令，可以测试运动卡通的当前坐标值，其语句格式如下：

XPOS(n)　　测试运动卡通的 X 坐标值

YPOS(n)　　测试运动卡通的 Y 坐标值

语句中的参数“n”为被测试运动卡通的序号，测试出的坐标取值范围：

X　为 0～255

Y　为 0～239

(二)优化程序的执行过程

程序 No. 4－9 运行后，首先执行 30、40 语句行，打印“玛丽哥哥走八边形”的拼音标题；50 行的作用是对玛丽运动起点坐标赋值；60～120 行完成玛丽走八边形的运动控制，其中 100 行用于测试当前运动是否完成，110 行用于测试玛丽一个方向的运动完成后所处位置的坐标，以作为下一个方向运动的起点；循环语句完成八个方向的行走变换控制。

第六节　服从命令听指挥

以上讨论的卡通都是自动运动的，我们只能在程序中对卡通的运动进行干预，一旦程序运行我们就束手无策了。而游戏中的卡通是在操纵器(游戏棒)的控制下运动的，人们可以随意控制卡通的前进、后退、跳跃、蹲伏等动作，F BASIC 能否实现这些控制呢？

一、操纵器上的按钮

电脑学习机上共有两个操纵器插孔，可插入两只操纵器，两只操纵器有主、副之分，一般把按钮多的一只称为主操纵器(编号为 0＃操纵器)，另一只为副操纵器(编号为 1＃)。它们的按钮排列和名称如图 4－5 所示。

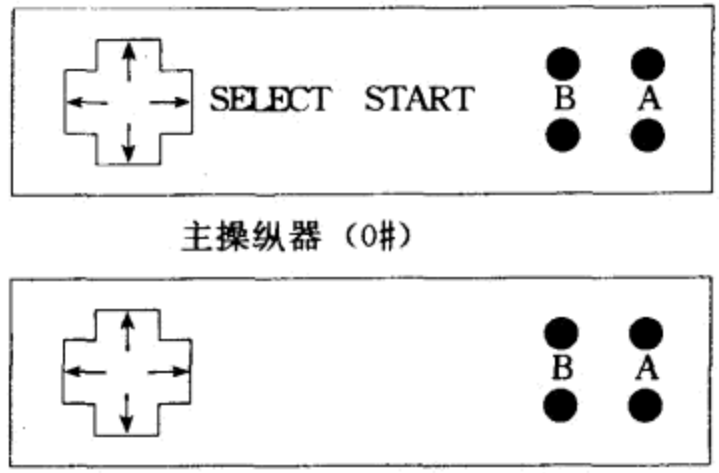


图 4—5　操纵器按钮分布图

1. **方向按钮**　操纵器左端的十字形按钮称为方向按钮。如同卡通运动方向一样，它可以取八个方向，游戏中一般由它控制卡通的运动方向。

2. **系统按钮**　主操纵器中的两个方形按钮“SELECT”、“START”称为系统按钮。游戏中一般由“SELECT”钮控制选择关次、武器、人数等；“START”钮用于控制游戏的开始、暂停等。系统按钮只有主操纵器配备。

3. **功能按钮**　操纵器中的圆形按钮称为功能按钮。游戏中一般由它们控制卡通的发射、跳跃等动作。功能按钮分上下两排，下排为单发钮（即按一下发射一枪或跳跃一次）；上排为连发钮（按住可连续跳跃或连续发射）。具体哪一个钮控制发射或跳跃，不同的游戏有不同的定义。

二、操纵器的控制指令

F BASIC 编程中，把操纵器上的按钮分为两类：方向按钮和功能按钮（系统按钮和功能按钮合为一类），分别用

“STICK”、“STRIG”两个函数代表它们的工作状态。函数定义如下：

1. **方向按钮的测试**　方向按钮工作状态的测试函数为“STICK”，其语句格式为

STICK(n)

参数“n”的取值由被测试操纵器确定，主操纵器取“0”，副操纵器取“1”。方向按钮各方向的函数值如图 4—6 括号所示。

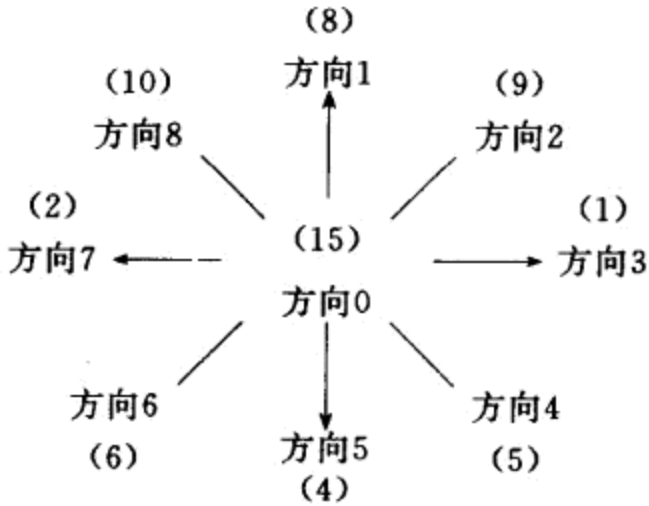


图 4—6　操纵器方向按钮函数值

由图 4—6 可见，斜方向的函数值其实就是两个正方向函数值的“和”。如“方向 8”(左上方向)的函数值“10”就是“方向左、方向上”函数值“2、8”相加得来的；“方向 0”的函数值则是“上、下、左、右”四个方向的函数值“8、4、2、1”相加得来的。

2. **功能按钮的测试**　功能按钮工作状态的测试函数为“STRIG”，其语句格式为

STRIG(n)

参数“n”的取值由被测试操纵器确定，主操纵器取“0”，副操

纵器取“1”。各功能按钮的函数值如表 4－3 所示。

表 4－3　操纵器功能按钮函数值对照表

按　　钮	函数值
开始暂停钮(START)	1
选择钮　　(SELECT)	2
单、连发钮(A)	8
单、连发钮(B)	4

三、操纵器函数演示程序

1. STICK 函数　操纵器方向按钮值的演示程序见 No. 4－10。

```
10 REM No. 4－10
20 CLS
30 S=STICK(0)
40 IF S=1 THEN PRINT "RIGHT"          (右)
50 IF S=2 THEN PRINT "LEFT"           (左)
60 IF S=4 THEN PRINT "DOWN"           (下)
70 IF S=8 THEN PRINT "UP"             (上)
80 IF S=5 THEN PRINT "DOWNRIGHT"(右下)
90 IF S=6 THEN PRINT "DEWNLEFT"  (左下)
100 IF S=9 THEN PRINT "UPRIGHT"   (右上)
110 IF S=10 THEN PRINT "UPLEFT"   (左上)
120 IF S=15 THEN PRINT "TOTAL"    (全部)
130 GOTO 30
```

演示程序选择了主操纵器，程序中括号中的内容为说明。

运行程序，使用主操纵器，当你按哪一个方向钮时，将显示该按钮的方向。

2. **STRIG 函数**　操纵器功能按钮值的演示程序见 No. 4－11。

```
10 REM No. 4－11
20 CLS
30 T＝STRIG(0)
40 IF T＝1 THEN PRINT "START"
50 IF T＝2 THEN PRINT "SELECT"
60 IF T＝4 THEN PRINT "B"
70 IF T＝8 THEN PRINT "A"
80 GOTO 30
```

演示程序选择了主操纵器。

运行程序，使用主操纵器，当你按哪一个功能钮时，将显示该按钮的名字。

四、听话的飞鹰

现在给大家一个使用主操纵器控制飞鸟运动的程序，供参考。见程序 No. 4－12。

```
10 REM No. 4－12
20 CLS: SPRITE ON: PRINT "PRESS START...": CLS
30 IF STRIG(0)<>1 THEN 30
40 X＝35: Y＝130: I＝1: GOSUB 200
50 K＝STICK(0): IF K＝0 THEN 50
60 IF K＝1 THEN I＝3
70 IF K＝2 THEN I＝7
80 IF K＝4 THEN I＝5
90 IF K＝5 THEN I＝4
```

```
100 IF K=6 THEN I=6
110 IF K=8 THEN I=1
120 IF K=9 THEN I=2
130 IF K=10 THEN I=8
140 X=XPOS(0): Y=YPOS(0)
150 GOSUB 200
160 GOTO 50
200 DEF MOVE(0)=SPRITE(15,I,1,4,0,0)
210 POSITION 0,X,Y:MOVE 0
220 IF MOVE(0)=-1 THEN 220
230 RETURN
```

运行程序后，必须按“START”钮，程序才向下运行。这时可使用主操纵器的方向钮控制飞鸟向八个方向飞翔。

五、飞鹰程序的优化

程序 No. 4－12 中，60～130 行共八行语句用于根据操纵器的方向函数值对飞鸟运动方向变量“I”赋值，程序显得较松散，能否把程序编得紧凑些呢？

使用 F BASIC 的关系运算符型函数可使程序简化。

关系运算符函数的使用格式如下：

（<变量><关系运算符><常量或变量>）

当使用关系运算符连接的表达式用括号括起后，就组成了关系运算符函数。当括号中的关系式成立时其函数值为“－1”，不成立时函数值等于“0”。如

$$(X=2)=\begin{cases}-1 & \text{当 } X=2 \text{ 时}\\ 0 & \text{当 } X\neq 2 \text{ 时}\end{cases}$$

使用关系运算符函数可使程序 No. 4－12 优化，优化后的程序见 No. 4－13。

```
10 REM No. 4－13
20 CLS：SPRITE ON:PRINT "PRESS START..."
30 IF STRIG(0)<>1 THEN 30
40 CLS：X＝35：Y＝130：I＝1：GOSUB 200
50 K＝STICK(0)：IF K＝0 THEN 50
60 I＝－(K＝1)＊3－(K＝2)＊7－(K＝4)＊5－(K＝5)＊4－(K
   ＝6)＊6－(K＝8)－(K＝9)＊2－(K＝10)＊8
70 X＝XPOS(0)：Y＝YPOS(0)
80 GOSUB 200：GOTO 50
200 DEF MOVE(0)＝SPRITE(15,I,1,4,0,0)
210 POSITION 0,X,Y:MOVE 0
220 IF MOVE(0)＝－1 THEN 220
230 RETURN
```

程序中的 60 行完成了对飞鸟运动方向变量“I”赋值的操作，由于方向函数变量“K”每一次只有一个确定的数值，故 60 行也必然对“I”赋以确定的方向值。

六、咒语困飞鹰

F BASIC 中还有两条控制运动卡通动作的指令：令运动卡通停止运动的指令“CUT”和令运动卡通消失的指令“ERA”。语句格式如下：

1. CUT **指令**　CUT 指令的语句格式为

```
CUTn0,[n1,n2,n3,n4,n5,n6,n7]
```

参数“n”为运动卡通的序号。CUT 指令一次可以令一个或数个(最多 8 个)运动卡通停止运动。

2. ERA **指令**　ERA 指令的语句格式为

```
ERA n0,[n1,n2,n3,n4,n5,n6,n7]
```

参数“n”为运动卡通的序号。ERA 指令一次可以令一个或数

个(最多 8 个)运动卡通消失。

现在改写程序 No. 4－13,要求:

用按钮"B"控制飞鸟停止运动(即咒语设困);

用按钮"A"控制飞鸟重新运动(解除咒语);

同时按"A、B"钮则令飞鸟消失;

再按方向钮令飞鸟重新显示并运动。

改写后的程序见 No. 4－14。

```
10 REM No. 4－14
20 CLS: SPRITE ON:PRINT "PRESS START..."
30 IF STRIG(0)<>1 THEN 30
40 CLS: X=35: Y=130: I=1: GOSUB 200
50 K=STICK(0): T=STRIG(0)
60 IF T=12 THEN ERA 0
70 IF T=4 THEN G=1: CUT 0
80 IF T=8 THEN G=0
90 IF G=1 OR K=0 THEN 50
100 I=-(K=1)*3-(K=2)*7-(K=4)*5-(K=5)*4-
    (K=6)*6-(K=8)-(K=9)*2-(K=10)*8
110 X=XPOS(0): Y=YPOS(0)
120 GOSUB 200: GOTO 50
200 DEF MOVE(0)=SPRITE(15,I,1,4,0,0)
210 POSITION 0,X,Y: MOVE 0
220 IF MOVE(0)=-1 THEN 220
230 RETURN
```

运行程序,飞鸟的运行控制同程序 No. 4－13 相同。按"B"钮后飞鸟停止运动,这时无论怎样按方向钮,飞鸟仍静止不动,如同咒语附身;只有按"A"钮时,飞鸟才重获自由;当同时按"A、B"钮时,飞鸟消失,只有按方向钮后飞鸟才重新显示

并在方向钮的控制下飞行。

飞鸟的停动、消失控制程序为 60～90 语句，结构较简单，留给读者分析。

七、卡通家族大聚会

F BASIC 共有 16 类卡通，能否令它们显示于一屏呢？

我们知道，F BASIC 允许一次定义八个显示卡通和八个运动卡通，这样，只要合理地安排程序就可以把 16 类卡通调出显示。程序见 No. 4－16。

```
10 REM No. 4－16
20 CLS: SPRITE ON
30 Y=－1
40 FOR I=148 TO 204 STEP 8
50 Y=Y+1
60 DEF SPRITE Y,(Y/2,1,0,0,0)=CHR$(I)+CHR$(I+1)
   +CHR$(I+2)+CHR$(I+3)
70 SPRITE Y,50*(1－Y*(Y<4)－(Y－4)*(Y>3)),50*(1
   +Y/4)
80 NEXT
90 FOR I=1 TO 7
100 DEF MOVE(I)=SPRITE(I,I+1,1,1,0,I/2)
110 POSITION I,50*(1－I*(I<4)－(I－4)*(I>3)),I/4*50
    +150
120 MOVE I: NEXT
```

运行程序，16 类卡通将以每行 4 个列队，排成一个 4×4 的方阵。

若能让这 16 类卡通满屏幕运动起来，将更为壮观。程序的改写留给读者练习。

第七节　本章小结

一、F BASIC 的卡通

F BASIC 中有两类卡通：显示卡通、运动卡通，每类卡通各允许定义 8 个，但在同一水平坐标上定义的卡通数量不允许超过 4 个，若超过 4 个时，后 4 个卡通将不能显示。

二、作用于卡通的指令

本章共讨论了与卡通有关的指令 13 条，分别为："SPRITE ON、SPRITE OFF、DEF SPRITE、SPRITE n，X，Y、SPRITE n、DEF MOVE、POSITION、MOVE、MOVE (n)、CUT、ERA、XPOS、YPOS"，现分别总结如下：

1. **打开卡通页显示指令**(SPRITE ON)

格式：SPRITE ON

类型：语句或命令

简写：SP. O.

功用：打开卡通页显示开关，使卡通页与背景页重叠显示。

注释：SPRITE ON 是打开卡通显示开关的意思。

SPRITE ON 为卡通在屏幕上显示提供了可能条件，至于卡通是否显示、显示哪种姿势、在何处显示还要取决于其它语句。

在执行 DEF SPRITE 或 DEF MOVE 语句之前，必须首先执行 SPRITE ON。否则，卡通将不能在画面上显示。

SPRITE ON 指令执行后，一直到执行 SPRITE OFF 指

令之前都有效。

2. **关闭卡通页显示指令**(SPRITE OFF)

格式:SPRITE OFF

类型:语句或命令

简写:SP. OF.

功用:关闭卡通页显示开关。

SPRITE OFF 指令可令显示于屏幕上的所有卡通消失。

注释:SPRITE OFF 为关闭卡通显示开关的意思。

3. **定义显示卡通指令**(DEF SPRITE)

格式:DEF SPRITEn,(A,B,C,D,E)=<字符表达式>

类型:语句

简写:DE. SP.

功用:选定显示于卡通页面上的卡通。

注释:DEF SPRITE 源于 DEFine a SPRITE,是定义一个显示卡通的意思。

语句格式中有六个参数,含义如下:

(1)n 是显示卡通的序号。

(2)A 为卡通配色号码。

(3)B 为卡通的大小。

(4)C 表示卡通的优先级。

(5)D 是显示卡通在 X 轴方向的翻转指示。

(6)E 是显示卡通在 Y 轴方向的翻转指示。

(7)字符表达式有两种形式:一种使用 CHR $ 指令组成,如"CHR $ (0)+CHR $ (1)+CHR $ (2)+CHR $ (3)"(定义玛丽走 1);另一种形式可用字符表示,如"DEFG"(定义飞鱼左 2)。

(8)六个参数都可以是常量、变量和表达式。六个参数的

具体使用要求请见正文的介绍。

4. **显示卡通的显示定位指令**(SPRITE n,X,Y)

格式:SPRITE n,X,Y

类型:语句

简写:SP. n,X,Y

功用:将由 DEF SPRITE 语句定义的显示卡通显示于任意位置。

注释:该语句将把由 DEF SPRITE 语句定义的 n 号显示卡通显示在(X,Y)位置上。

n——为 DEF SPRITE 语句定义的卡通序号。

X——为卡通页面上的水平坐标,取值范围 0～255。

Y——为卡通页面上的垂直坐标,取值范围 0～255。

n、X、Y 均可取常数、变量或表达式。

说明:

(1)由于电视机屏幕的差异,实际显示的卡通面往往达不到 255×255 大小,若以 255 取值,可能会出现卡通显示不出来的情况。因此,编程中建议选取水平坐标的取值范围 0～240,垂直坐标的取值范围 0～220。

(2)SPRITE n,X,Y 语句可在一个画面上同时显示 8 个显示卡通,超过 8 个显示卡通时,超过的部分将不能显示。

(3)卡通在同一水平方向上只能显示 4 个卡通,超过 4 个以上的将不能显示。

(4)若在同一坐标上显示多个卡通,编号小的卡通将显示在最前面,即编号小的卡通显示优先度最大。

(5)由于在 DEF SPRITE 语句中,B=1 时,卡通将有 4 个字符大小,故由 SPRITE n,X,Y 语句决定的显示坐标将作用于卡通的左上角。

5. **隐藏显示卡通的指令**(SPRITE n)

格式:SPRITE n

类型:语句

简写:SP. n

功用:令已显示的显示卡通在画面上消失。

注释:n 为由 DEF SPRITE 语句定义的显示卡通编号,取值范围 0～7。它可以是常数、变量或表达式。

6. **运动卡通定义指令**(DEF MOVE)

格式:DEF MOVE(n)=SPRITE(A,B,C,D,E,F)

类型:语句

简写:DE. M.

功用:定义运动卡通。

通过"DEF MOVE"语句确定运动卡通的编号、运动卡通的种类、运动方向、运动速度、一次运动距离、相对于背景的显示优先度和运动卡通的配色。

注释:DEF MOVE 取自"DEFine MOVE",定义运动的意思。

n——运动卡通的编号,取值 0～7。

A——运动卡通的种类编号,取值 0～15。

B——运动卡通的运动方向,取值 0～8。

C——运动卡通的运动速度,取值 1～255。

D——运动卡通的一次运动量,取值 1～255。

E——运动卡通的显示优先度,取值 0、1。

F——运动卡通的配色号码,取值 0～3。

以上七个参数可以取常数、变量或表达式。

说明:

在 DEF MOVE 语句中运动方向(B)、显示优先度(E)、

配色号码(F)三个参数可以省略。省略B后，系统自动选定运动方向为4；省略E时则自动设定E＝0；省略F时则自动设定F＝0，并使运动卡通从(0,0)坐标处开始运动。省略中要注意，参数可以省略但参数分隔符要保留。如

DEF MOVE(0)＝SPRITE(2,,1,20,,)

若同时省略E、F时，可把最后的分隔符去掉。如上述语句也可以写为

DEF MOVE(0)＝SPRITE(2,,1,20)

7. **运动卡通定位指令**(POSITION)

格式：POSITION n,X,Y

类型：语句

简写：POS.

功用：规定运动卡通运动前的坐标。

注释："POSITION"是位置、定位的意思。

n——由DEF MOVE语句定义的运动卡通编号，取值范围0～7。

X——规定运动卡通运动前的水平坐标，取值范围0～255。

Y——规定运动卡通运动前的垂直坐标，取值范围0～255。

当编程中未指定运动卡通的运动起始坐标时，系统自动约定为X＝120，Y＝120。

三个参数可以是常量、变量或表达式。

8. **运动卡通启动指令**(MOVE)

格式：MOVE n0[,n1,n2,n3,n4,n5,n6,n7]

类型：语句

简写：M.

功用:启动运动卡通开始运动。

注释:"MOVE"为动作、运动的意思。

n0～n7 为由 DEF MOVE 语句定义的运动卡通编号。

MOVE 语句可以一次启动 8 个运动卡通开始运动,也可以只启动一个。如

MOVE 1

MOVE 1,3,5,7

MOVE 7,3,5,1,4,0

等都是合法的。

9. **运动卡通运动完成测试指令**(MOVE(n))

格式:MOVE(n)

类型:函数

简写:M.(n)

功用:测试运动卡通的当前运动是否完成。

注释:n 是由 DEFMOVE 语句定义的运动卡通编号,取值范围 0～7。

当定义的运动卡通经 MOVE 语句启动开始运动后,MOVE(n)函数将由指定序号的卡通动作赋给新的数值:

若在运动中,函数值为"－1";

若运动完成函数值为"0"。

10. **运动卡通停止指令**(CUT)

格式:CUT n0[,n1,n2,n3,n4,n5,n6,n7]

类型:语句

简写:CU.

功用:令由 MOVE 语句启动的运动卡通停止运动。

CUT 语句停止运动的卡通,可用 MOVE 语句再启动。再启动后,卡通将继续执行停止运动前未执行完的运动量,直到

执行完定义的运动量后才停止。

注释:"CUT"取自"CUTmove",是停止运动的意思。

n0～n7 是由 DEFMOVE 语句定义的运动卡通编号。CUT 语句可以一次停止 8 个运动卡通的动作,也可以只停止一个卡通。如:

```
CUT1
CUT0,1,3,7
CUT7,4,1,3,6
```

等都是合法的。

11. **运动卡通消失指令**(ERA)

格式:ERA n0[,n1,n2,n3,n4,n5,n6,n7]

类型:语句

简写:ER.

功用:令由 MOVE 语句启动的运动卡通停止运动,并从画面上消失。

ERA 语句还可以令由 CUT 语句停止运动的卡通消失。

被 ERA 语句作用消失的卡通,可用 MOVE 语句令其重新显示,并继续运动消失前未执行完的运动量。重新显示的位置坐标仍是消失前的坐标。

注释:"ERA"取自"ERAse",是擦除的意思。

n0～n7 是由 DEF MOVE 语句定义的运动卡通编号。ERA 语句可以一次停止并消失 8 个运动卡通,也可以只消失一个卡通。如:

```
ERA1
ERA0,1,3,7
ERA7,4,1,3,6
```

等都是合法的。

12. **测试运动卡通水平坐标指令**(XPOS)

格式:XPOS(n)

类型:函数

简写:XP.

功用:求出运动卡通当前的水平坐标值。

注释:"XPOS"取自于"X axis POSition",为沿 X 轴位置的意思。

n 为 DEF MOVE 语句定义的运动卡通编号,取值范围 0～7,可以是变量或表达式。

13. **测试运动卡通垂直坐标指令**(YPOS)

格式:YPOS(n)

类型:函数

简写:YP.

功用:求出运动卡通当前的垂直坐标值。

注释:"YPOS"取自于"Y axis POSition",为沿 Y 轴位置的意思。

n 为 DEF MOVE 语句定义的运动卡通编号,取值范围 0～7,可以是变量或表达式。

三、其它指令

本章还介绍了操纵器测试指令 STICK、STRIG,延时控制指令 PAUSE,光标定位指令 LOCATE 和 ASCII 码——字符转换指令 CHR $。

1. **操纵器方向钮测试指令**(STICK)

格式:STICK(m)

类型:函数

简写:STI.

功用:测试操纵器方向钮的工作状态。

操纵器上的每个方向钮按下,都将由 STICK 函数返还一个确定的函数值,没有钮按下时其函数值为"0"。

注释:"STICK"是指挥棒、操纵杆的意思。

m 为被测试操纵器的代码,取值 0、1。m=0 表示测试主操纵器(1 号操纵器);m=1 测试副操纵器(2 号操纵器)。m 可以取常数、变量或表达式。

操纵器各方向按钮的函数值请参见本章第五节。

2. **操纵器功能钮测试指令**(STRIG)

格式:STRIG(m)

类型:函数

简写:STRI.

功用:测试操纵器功能钮 START、SELECT、A、B 的工作状态。

操纵器上的每个功能钮按下,都将由 STRIG 函数返还一个确定的函数值,没有钮按下时其函数值为"0"。

注释:"STRIG"取自"STRIGger",为"触发"的意思。

m 为被测试操纵器的代码,取值 0、1。m=0 表示测试主操纵器(1 号操纵器);m=1 测试副操纵器(2 号操纵器)。m 可以取常数、变量或表达式。

操纵器各功能按钮的函数值请参见本章第五节。

3. **延时控制指令**(PAUSE)

格式:PAUSE<数值>

类型:语句

简写:PA.

功用:令程序运行暂停一段时间后再继续运行。

PAUSE 语句主要用于程序运行中的延时或暂停。

注释:"PAUSE"是"中止或暂停"的意思。

PAUSE 语句中的数值为暂停时间或延时量。取值范围为 0～32767。

若"PAUSE"后面不带数值,当程序执行到该语句时将暂停,出现一闪烁光标,等待用户按任意键后再向下执行。

4. **光标定位指令**(LOCATE)

格式:LOCATEX,Y

类型:语句或命令

简写:LOC.

功用:在程序的执行期间对光标进行定位控制。

LOCATE 语句为用户在程序执行期间对光标进行各种控制提供了有效的手段。它可以对输出到屏幕上的数据或图形进行定位显示。

注释:"LOCATE"在英文中为"定位、放置"的意思。

当 LOCATE 语句被执行后,光标便被定位到行与列所确定的交点上。行和列用一对数值表示,可以取变量形式,也可以是表达式。X 的取值范围为 0～27,Y 的取值范围为 0～23。

示例:编写打印菱形的程序

```
10 REM No. 4－17
20 CLS
30 FOR I=0 TO 10
40 LOCATE 13+I, I:PRINT " $"
50 LOCATE 13－I, I:PRINT " $"
60 LOCATE 13+I, 20－I: PRINT " $"
70 LOCATE 13－I, 20－I: PRINT " $"
80 NEXT: LOCATE 0,0
```

运行程序，将打印出如图 4－7 所示的菱形图案。

图 4－7　菱形图案

注意——使用 LOCATE 语句设定闪烁光标的位置时，X 的取值恒为“0”，即无论 X 取何值，电脑一律把 X 当做“0”处理，这是因为闪烁光标为输入语句的起点，而每行语句的输入起点必须在 X＝0 的位置，所以

LOCATE 0,15

和

LOCATE 10,15

的执行效果是一样的。

5. ASCII **码——字符转换指令**(CHR＄)

格式：CHR＄(X)

类型：函数

简写：CH.

功用：将某个十进制数的 ASCII 码转换为对应的字符。

注释：“CHR＄”取自于“CHaRacter”，是“字符”的意思。后面又加了美元号(＄)，意思是把 ASCII 码换为字符。“＄”

在这里形象地表示“string”，即“串”。BASIC 语言中对与字符串操作有关的函数都加上 $ 。

X 是一个 ASCII 代码，CHR $ 函数生成的是这个代码所对应的字符。X 取值范围为 0～255 的整数，它可以是常数、变量或表达式。

第五章
七彩缤纷的画面

本章重点讨论 F BASIC 的配色系统，并结合绘图介绍配色、字符串处理等指令：

随机数函数指令	RND
屏幕染色指令	COLOR
调色板设定指令	CGSET
背景配色代码设置指令	PALETB
卡通配色代码设置指令	PALETS
从字串左边取子串指令	LEFT $
从字串右边取子串指令	RIGHT $
从字串的任意位置取子串指令	MID $
字符串长度测试指令	LEN
求绝对值指令	ABS

第一节　屏幕万花筒

用三块长方形的玻璃围成三棱体，外边用不透光的黑纸包裹，把三棱体的一端用半透明的物体（纸、玻璃等）遮住，剪一些各种颜色的碎纸片放入三棱体内就构成了一个简易的万花筒。把万花筒对着明亮处，从另一端观看，同时缓慢转动万花筒，将看到变化莫测、绚丽多彩的立体画面。

能否让电脑在屏幕上模拟出万花筒的效果呢？

一、屏幕染色示范程序

请运行程序 No. 5－1。

```
10 REM No. 5－1 PING MU WAN HUA TONG
20 CLS: FOR I=0 TO 671: PRINT CHR $ (255);: NEXT
30 X=RND(28): Y=RND(24): Z=RND(4)
40 COLOR X,Y,Z
50 GOTO 30
```

运行程序，首先打印出 28 列×24 行的一个蓝色矩形，然后在蓝色的矩形上开始显示变幻莫测的黄、红、绿色块，很快各种色块布满屏幕，并一直变化不停，形成效果非凡的屏幕万花筒。

二、随机数的产生

30 行中的“RND”是随机数产生函数，语句格式为

RND(X)

功能是随机产生 0～X 之间的一个整数。如

RND(20)

RND(300)

等。请见如下示例程序

```
10 CLS
20 PRINT RND(20);
30 GOTO 20
```

运行程序，将打印出

1 19 14 17 4 3 6 16 13 2 4

14 19 14 1 9 15 ……

打印出的数据都是 0～20 之间的整数。

三、屏幕的染色

程序 No. 5－1 中还出现了一个新指令“COLOR”，“COLOR”为屏幕染色指令，其语句格式为

COLOR X,Y,n

功能是将屏幕上坐标为(X,Y)处的背景图案，染上“n”号配色代码。

在 COLOR 语句的作用下，可对屏幕任意位置的背景图案染色，染色位置由 X、Y 决定。

X 为屏幕水平方向的位置坐标，取值范围 0～27；

Y 为垂直方向的位置坐标，取值范围 0～23

n 为选定的配色代码，取值 0～3。

四、让万花筒更绚丽

程序 No. 5－1 的 20 行使用了代码为 255 的背景图案，由封二可见该图案是单一深蓝色的方块。因此，由程序 20 行控制画出了一块单色的大矩形。若选用背景图形中代码为 253、254、255 三类色块，随机组合成大矩形，那么，该矩形本身将有三种色彩的变化。这时再使用 COLOR 语句染色，将使屏幕万花筒的色彩更绚丽。程序见 No. 5－2。

```
10 REM No. 5－2 PING MU WAN HUA TONG
20 CLS: FOR I=0 TO 671: PRINT CHR$(RND(3)+253);:
   NEXT
30 X=RND(28): Y=RND(24): Z=RND(4)
40 COLOR X,Y,Z
50 GOTO 30
```

程序中 20 行的

CHR $ (RND(3)+253)

随机函数产生 0、1、2 三个随机数，再加 253 后，将合成 253、254、255 三个数据，因而随机选择了背景图案中的最后三种图形。

运行程序 No. 5－2，将看到更精彩的屏幕万花筒。

若将 20 行改为

```
20 CLS: FOR I=0 TO 671: PRINT CHR $ (RND(71)+184);:
   NEXT
```

将打印出更复杂的画面，显示更丰富的色彩。

若在程序 No. 5－2 中加入 35 行程序

```
35 CGSET RND(2),1
```

会变成什么样子呢？

请运行程序试一试……。

哇！真精彩！

屏幕上不但某个色块的颜色在随机变化，而且整个画面中各个色块的颜色也在随机变化，并产生了闪光的效果。

显然是 35 行中的“CGSET”指令在妙笔生花。

妙笔是如何生花的呢？

这要从 F BASIC 的颜色谈起……。

第二节　诱人的金山

本节重点讨论 F BASIC 的调色系统，通过绘制金山程序的编写，展现 F BASIC 调色系统的奇妙。

一、F BASIC 的颜色

众所周知，绘制油画需要油彩，油彩有红、绿、白、黑等多

种颜色，每一种颜色中又有多种过渡色，如红色有桃红、深红、玫瑰红等。同样，电脑设计彩色卡通或背景也需[illegible]、使用各种颜色。F BASIC 使用了 61 种颜色，每一种颜色规定了一个代号，共有 61 个代号，它们的对应关系见表 5－1。

表 5－1　61 种颜色代码表

行别	系别	0 列		1 列		2 列		3 列	
		十六进制	十进制	十六进制	十进制	十六进制	十进制	十六进制	十进制
0	灰白系	00	0	10	16	20	32	30	48n
0	蓝系	01	1	11	17	21	33	31	49
1		02	2	12	18	22	34	32	50
2		03	3	13	19	23	35	33	51
3		04	4	14	20	24	36	34	52
0	红系	05	5	15	21	25	37	35	53
1		06	6	16	22	26	38	36	54
2		07	7	17	23	27	39	37	55
3		08	8	18	24	28	40	38	56
0	绿系	09	9	19	25	29	41	39	57
1		0A	10	1A	26	2A	42	3A	58
2		0B	11	1B	27	2B	43	3B	59
3		0C	12	1C	28	2C	44	3C	60
0	黑系	0D	13	1D	29	2D	45		
1		0E	14	1E	30	2E	46		
2		0F	15	1F	31	2F	47		

暗←——————————→明

表 5－1 中列出了颜色代码的 16 进制、10 进制两种形

式，由上至下颜色共分 5 个系列：灰白颜色系列有 4 种颜色；蓝色、红色、绿色系列各有 16 种颜色；黑色系列有 9 种颜色。由左向右颜色的亮度分为 4 列，列号越大越明亮，色调则越来越淡。蓝、红、绿三个色系的 16 种颜色各分为 4 行，其中间的两行颜色较纯正，第一、第四行的色调则向其相邻色系靠近，如红色系的第一行颜色为玫瑰红（颜色偏紫），第四行的颜色则为赭红（颜色偏绿）。

在表 5－1 中虽然每一种颜色有一个确定的代码，但对于给定的颜色代码，若不查表一时不能确定是何种颜色。因此，为了区分一个色系中的各种颜色，我们在表 5－1 中安排了列号和行号。这样，每种颜色代码可由其所在的列号和行号唯一确定，如代码为 36 的颜色列号为“2”、行号为“3”，可称为“蓝 23”，即蓝 23 的颜色代码为 36。这样表示可马上知道该颜色代码是哪一个色系的颜色。

二、配色代码

我们知道，F BASIC 中有背景和卡通两类图案，每类图案都有不同的颜色组合，如玛丽显示蓝色、丽莎显示红色（在配色号码等于 0 时），当然也可以通过改变配色代码让他们显示其它颜色。

在以上定义卡通的讨论中，我们只知道改变配色代码的取值可以改变卡通的颜色，而且还知道配色代码的取值范围为 0～3，其它就一无所知了。

那么，配色代码是什么呢？

原来，在设计卡通或背景图案时，F BASIC 允许每一个图案选用三种颜色进行绘制。也就是说选三种颜色组成一个调色盘，如选颜色代码“1（深蓝）、22（深红）、48（纯白）”三种颜

色绘制玛丽，就绘出了玛丽走 1 的图案。

调色盘可以有很多个，为了区分，人们对调色盘进行编号。这个编号就叫做配色代码。

显然，一个配色代码代表 3 种颜色。

三、调色板

若只用一个配色代码的三种颜色绘制图案，画面太单调，若多组几个调色盘，选色的范围增大，将绘制出较生动的图案。F BASIC 允许在一屏画面中使用 4 个调色盘绘制图案，因而就产生了 4 个配色代码，编号分别为 0～3。我们把这 4 个调色盘(12 种颜色)称为一个调色组。它就如同在一个板上放置了 4 个调色盘一样，故称为“调色板”。

换言之，“调色板”就是由 4 个调色盘、12 种颜色构成的颜色组合。

F BASIC 允许背景图案有两个调色板，卡通图案有三个调色板。因此，背景调色板的编号为 0、1；卡通调色板的编号为 0、1、2。

显然，调色板的用途就是给背景和卡通图案配置颜色。

总结 F BASIC 的配色系统结构可简述如下：

由 3 个颜色代码组成一个配色代码；

由 4 个配色代码组成一个调色板代码；

由两个背景调色板代码组成背景调色系统；

由三个卡通调色板代码组成卡通调色系统。背景调色系统的结构示意如图 5—1 所示。

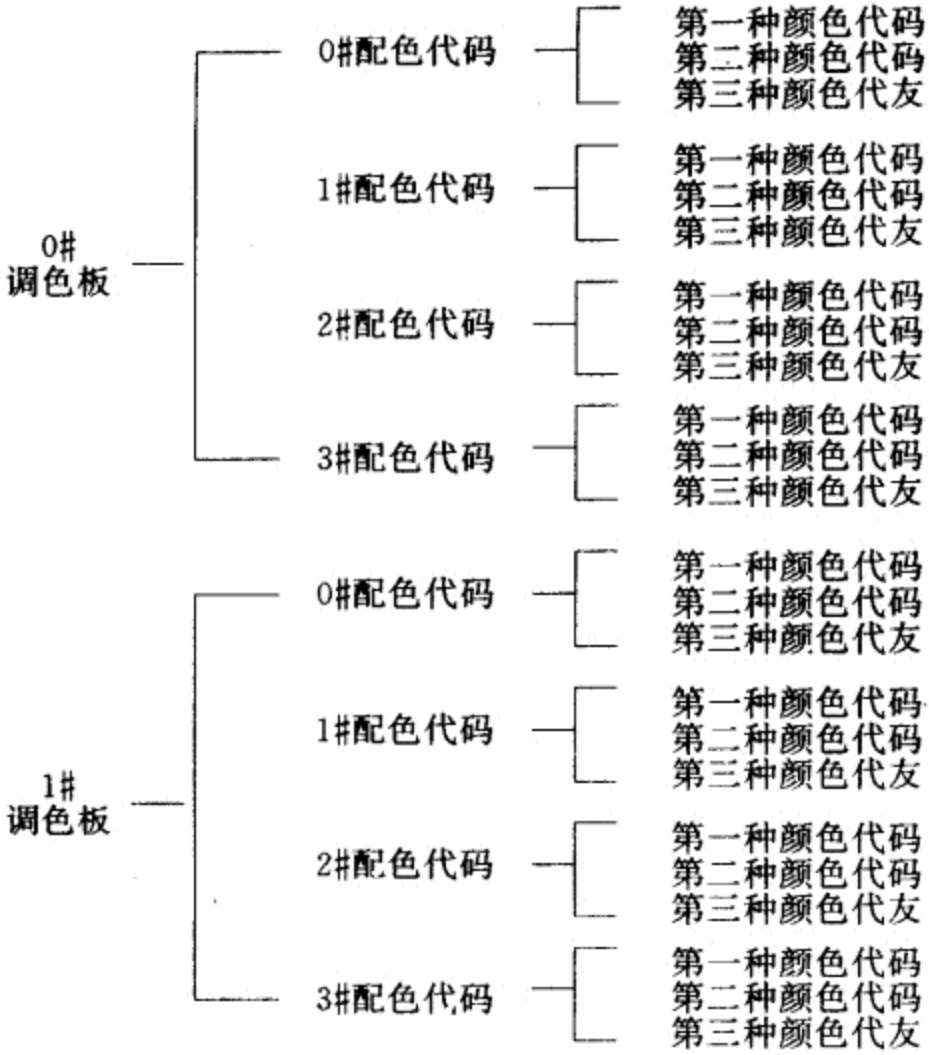


图 5—1　背景调色系统结构示意图

卡通调色系统除比背景多出一个调色板以外，其它部分结构相同。

四、 F BASIC 的调色系统数据

为方便使用，F BASIC 系统中已定义背景、卡通两套调色系统，它们的数据如下：

1. 背景调色系统数据

背景调色系统数据见表 5—2。

表 5—2　背景调色数据一览表

0#调色板

0#配色代码	绿 23	红 10	红 02
	44	21	7
	2CH	15H	07
1#配色代码	绿 30	蓝 32	蓝 11
	39	33	18
	27H	21H	12H
2#配色代码	绿 20	红 31	红 12
	41	54	23
	29H	36H	17H
3#配色代码	白 30	红 21	红 02
	48	38	7
	30H	26H	07

1#调色板

0#配色代码	白 30	蓝 20	蓝 01
	48	33	2
	30H	21H	02
1#配色代码	白 30	绿 30	红 13
	48	39	24
	30H	27H	18H
2#配色代码	白 30	绿 30	红 11
	48	39	22
	30H	27H	16H
3#配色代码	绿 20	红 31	红 12
	41	54	23
	29H	36H	17H

表中每种配色代码颜色的下面一行为颜色代码的十进制数，第二行为对应的十六进制数。

2. 卡通调色系统数据

卡通调色系统数据见表 5—3。

表 5—3　卡通调色系统数据一览表

0#调色板

0# 配色代码	红 31	红 11	蓝 01
	54	22	2
	36H	16H	02
1# 配色代码	红 32	白 30	绿 20
	55	18	41
	37H	30H	29H
2# 配色代码	红 30	红 20	红 12
	53	37	23
	35H	25H	17H
3# 配色代码	白 30	红 22	红 11
	48	39	22
	30H	27H	16H

1#调色板

0# 配色代码	白 30	红 11	蓝 00
	18	22	1
	30H	16H	01
1# 配色代码	白 10	白 00	蓝 00
	16	0	1
	10H	00	01
2# 配色代码	白 30	绿 20	绿 00
	48	41	9
	30H	29H	9
3# 配色代码	白 30	红 11	红 02
	48	22	7
	30H	16H	07

2#调色板

0# 配色代码	白 30	红 21	红 13
	18	38	18
	30H	26H	12H
1# 配色代码	白 30	红 10	蓝 11
	18	21	18
	30H	15H	12H
2# 配色代码	白 30	蓝 11	红 11
	48	18	22
	30H	12H	16H
3# 配色代码	白 30	红 21	红 10
	48	38	25
	30H	26H	19H

五、F BASIC 的染色坐标

我们知道染色语句“COLOR X,Y,n”可以对坐标为(X,Y)处的图形染上 n 号配色代码的颜色，但是 COLOR 指令的染色单位是以 4 个字符大小为单位的，即执行一次染色语句可以把对应坐标相邻的 4 个字符进行染色。我们必须了解染

色和坐标的对应关系，才能灵活地使用染色语句，达到预期的染色效果。

F BASIC 把屏幕的横向每两个字符为单位分成 14 等份，纵向除第一行和最后一行外每两行为单位分成 11 等份，使其构成了图 5－2 所示的 154 个 2×2 的方格和 28 个 1×2 的方格，COLOR 指令就作用在这些方格上。

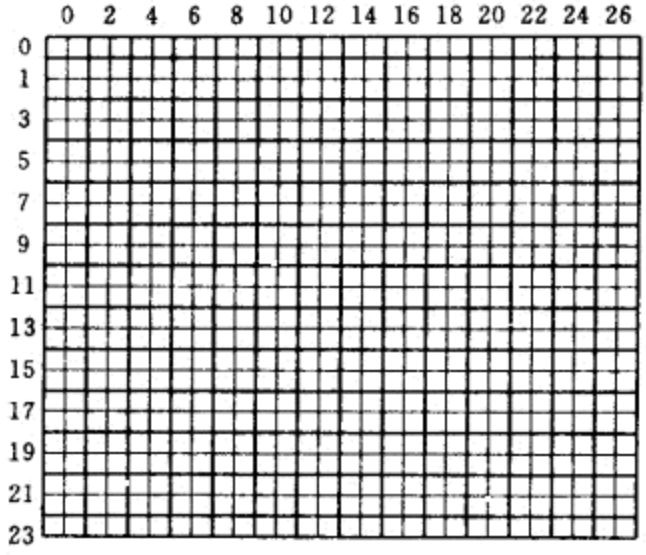


图 5－2　COLOR 指令染色坐标与屏幕对应图

图 5－2 表示了一个屏幕可按 4 格组成一个方块，使用 COLOR 指令指定坐标，可把方块内部的四块染成配色代码的颜色。如屏幕上的某块图形的坐标为

	10	11	
第 7 行	A	B	
第 8 行	C	D	

使用　　COLOR 10,7,3

COLOR 11,7,3

COLOR 10,8,3

COLOR 11,8,3

中任一条语句都可以将“A、B、C、D”4 块图形染成 3 号配色代码的颜色。这说明了 COLOR 指令作用的区域可从 5－2 中方便地查出。

由以上讨论可以得出 COLOR 语句的染色特点如下：

1. COLOR 语句只能对背景图案染色，不能对底背景染色。

2. COLOR 语句执行后，不仅对指定坐标的图形染色而且还把与其相邻的、处于同一方块中的三个图形染上同一颜色。

3. 由于 COLOR 语句一次对四个图形染色，所以它染色的色块较大。

为使大家加深对 COLOR 指令的认识，我们做一道例题。

例 5－1　用不同颜色矩形作金砖，垒起一座金山。要求金山上下七层，最下层七块、最上层一块，共使用 28 块金砖。

程序见 No. 5－3

```
10 REM No. 5－3
20 CLS: A$ =CHR$ (255): A$ =A$ +A$
30 FOR J=1 TO 7
40 X=(7－J) * 2
50 FOR I=1 TO J
60 LOCATE X,J * 2+3: PRINT A$
70 LOCATE X,J * 2+4: PRINT A$
80 COLOR X,J * 2+4,RND(4)
90 X=X+4: NEXT
```

```
100 NEXT
110 LOCATE 0,22
```

运行程序，打印出的金山见图 5－3。

图 5－3　用 COLOR 语句画出的金山

学习了 F BASIC 的配色系统后，我们知道，金砖的颜色是由 COLOR 语句通过改变配色代码而改变的。我们还知道，背景图案有两个调色板，那么，COLOR 语句选用的配色代码究竟是哪一个调色板上的呢？

这就用到了程序 No. 5－2 中出现的“CGSET”指令。

CGSET 指令的语句格式为：

CGSET＜m，n＞

语句中的“m”是选择背景调色板的参数，“n”为选择卡通调色板的参数。显然，“m”的取值为 0、1；“n”的取值为“0～2”。

程序 No. 5－2 的画面能够产生闪光效果，就是改变了背景调色板的缘故。

一般在程序的开始，首先编写 CGSET 语句以设定背景、卡通图案的调色板。

当复位后，系统自动设定为

CGSET 1，1

故当不执行该语句时，系统设定背景、卡通图形均使用编号为“1”的调色板。

了解了 CGSET 指令的功用后，我们再来改造程序 No. 5－3，优化后的程序见 No. 5－4。

```
10 REM No. 5－4
```

```
20 CLS: A$=CHR$(255): A$=A$+A$
30 FOR J=1 TO 7: X=(7-J)*2
40 FOR I=1 TO J
50 LOCATE X,J*2+3: PRINT A$
60 LOCATE X,J*2+4: PRINT A$
70 COLOR X,J*2+4,RND(4)
80 X=X+4: NEXT: NEXT
90 PAUSE 100: A=A+1: IF A>1 THEN A=0
100 CGSET A,1
110 GOTO 30
```

对比程序 No. 5－3，No. 5－4 的色彩有了更丰富的变化。

六、改变底背景和配色代码的颜色

F BASIC 还专门设置了改变配色代码中三种颜色组合和底背景颜色的指令“PALETB”、“PALETS”，其中“PALETB”用于改变背景图案配色代码的颜色组合，“PALETS”用于改变卡通图案配色代码的颜色组合。其语句格式如下：

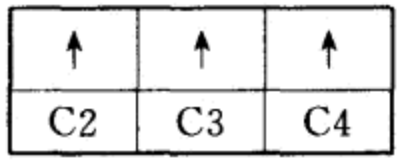
```
PALETB n,C1,C2,C3,C4
PALETS n,C1,C2,C3,C4
```

语句中的“n”为配色代码的序号，取值 0～3，它可以是常数、变量或表达式。

C1 用于指定底背景的颜色代码。开机后，底背景的颜色自动设定为黑色。当我们欲改变底背景的颜色时，可以通过改变 C1 的取值实现，C1 为表 5－1《61 种颜色代码》表中的任一种颜色代码。但在指定底背景颜色时，必须首先打开底背景置色开关。该开关通过改变配色代码“n ”的取值来控制，n＝0 打开开关；n≠0 则无效。C1 的取值范围 0～60，它可以取常

数、变量或表达式。

C2、C3、C4 分别用于指定任意三种颜色代码，它们分别对应每个配色代码中左、中、右的三个颜色代码。对应关系如下：



上表中的箭头分别表示 C2、C3、C4 所对应的颜色。C2、C3、C4 的取值范围为 0～60，也可以是常数、变量和表达式。

C1～C4 可以用十进制数表示，也可以用十六进制数表示。当取十六进制数形式时，要在十六进制数值的前面添“&H”。

示例：

PALETB 0,7,44,21,7

执行后把底背景置为暗红色。

PALETB 3,&H0F,&H30,&H26,&H12

执行后，把背景调色板中 3 号配色代码中的 C2、C3、C4 颜色代码换为“48、38、18”。

使用“PALETS”指令可以改变卡通配色代码的颜色组合，请见程序 No. 5－5。

```
10 REM No. 5－5 "PALETS"
20 CGSET 0,0: CLS
30 SPRITE ON
40 DEF MOVE(0)=SPRITE(0,3,1,120,0,0): MOVE 0
50 IF MOVE(0)=－1 THEN 50
60 PALETS 0,21,54,24,25: MOVE 0
```

运行程序，玛丽先以原配色(54,22,2)的颜色组合运动

(主色为蓝、红)；随后变为(54，24，25)的颜色组合运动(主色为绿、红)，同时底背景的颜色变为粉红色。

至于，使用“PALETB、PALETS”指令改变哪一个调色板中的配色代码，则由“CGSET”语句决定。

七、金山放异彩

学习了“PALETB”指令后，可对程序 No. 5－4 进行更进一步的改造，使金山大放异彩。改造后的程序见 No. 5－6。

```
10 REM No. 5－6
20 CLS：A$＝CHR$(255)：A$＝A$＋A$
30 FOR J＝1 TO 7：X＝(7－J)＊2
40 FOR I＝1 TO J
50 LOCATE X，J＊2＋3：PRINT A$
60 LOCATE X，J＊2＋4：PRINT A$
70 COLOR X，J＊2＋4，RND(4)
80 X＝X＋4：NEXT：NEXT
90 A＝A＋1：IF A＞1 THEN A＝0
100 CGSET A，1：PALETB 0，RND(12)＋1，RND(61)，RND
    (61)，RND(61)
110 PAUSE 5：GOTO 90
```

运行程序，在流金泛银的底背景画面上，61 种色彩变幻的金山渗透着灵性，令人眼花缭乱，彷佛此身已入金山中，人与金山共光辉。

八、完全对称的屏幕万花筒

在上一节编程打印的屏幕万花筒是杂乱无章的，现在介绍一个上下左右完全对称的屏幕万花筒程序。见 No. 5－7。

```
10 REM No. 5－7
```

```
20 CLS
30 C=RND(4): C$=CHR$(RND(71)+184): PALETS 0,
   RND(12)+1,0,0,0
40 X=RND(24)+1: Y=RND(20)+1
50 DX=RND(4): DY=RND(4): IF DX=0 OR DY=0 THEN
   20
60 IF X<1 OR X>27 OR Y<1 OR Y>22 THEN 20
70 LOCATE X,Y: PRINT C$: COLOR X,Y,C
80 LOCATE 28-X,Y: PRINT C$: COLOR 28-X,Y,C
90 LOCATE X,24-Y: PRINT C$: COLOR X,24-Y,C
100 LOCATE 28-X,24-Y: PRINT C$: COLOR 28-X,24-
    Y,C
110 X=X+DX: Y=Y+DY: GOTO 60
```

运行程序，屏幕上周而复始地打印出一个个对称的图案。

总之，F BASIC 的调色系统的结构较复杂。但只要注意分清颜色代码、配色代码、面板代码之间的层次关系和设置定义三类代码的有关指令，就可理出头绪，运用随心。

第三节　神奇的设计师

一、字符宝塔

例 5－2　编程打印如图 5－4 所示的 5 层字符宝塔。

```
    A
   AAA
  AAAAA
 AAAAAAA
AAAAAAAAA
```

图 5－4　字符宝塔

欲打印图 5－4 的宝塔，须使用双重循环，外层循环用于控制打印宝塔的五层，内层循环用于控制每层打印的字符数量。实现程序如 No. 5－8。

```
10 REM No. 5－8, BAO TA 1
20 CLS: FOR I=1 TO 6
30 LOCATE 15－I * 2,I * 2
40 FOR J=1 TO I * 2－1
50 PRINT "A";
60 NEXT: NEXT
```

运行程序将打印出图 5－4 所示的宝塔。

程序 No. 5－8 实际上可以打印任意层数的宝塔，但由于 F BASIC 屏幕显示的限制，最多可打印出 7 层宝塔。

其实，使用单循环也可以打印出字符宝塔，请见程序 No. 5－9。

```
10 REM No. 5－9, BAO TA 2
20 CLS: A$ ="ABCDEFGHIJK": A=0
30 FOR I=1 TO 11 STEP 2
40 A=A+1: LOCATE 13－A,5+A
50 PRINT LEFT$ (A$ ,I)
60 NEXT
```

运行程序，将打印出图 5－5 所示的宝塔。

```
A
ABC
ABCDE
ABCDEFG
ABCDEFGHI
ABCDEFGHIJK
```

图 5－5　字符宝塔

二、老子和儿子

1. 从字串左边取子串　程序 No. 5－9 中使用了一条新指令“LEFT $”，它是一个“从字符串左边取子字符串”的函数。子字符串的概念是：对于字符串：

A $ ="ABCDEFG"

来说，字符串

AB

ABC

ABCDE

BCDEF

CDEF

EF

等，都可以称为字符串 A $ 的子字符串，简称“子串”。它的语句形式为

LEFT $ (字符串，n)

语句的功用是从字符串左边开始取出 n 个字符形成一个子字符串。

程序 No. 5－9 的 30 行控制每次从字符串 A $ 中取出“I”个字符，由于“I”的取值为“1、3、5、7、9”，所以打印出字符宝塔。可见，程序 No. 5－9 所打印宝塔中每行的字符是可以选择的。因此，程序 No. 5－9 比 No. 5－8 的功能更强。

2. 从字串右边取子串

例 5－3　编程打印图 5－6 所示的菱形。

```
A
A B C
A B C D E
A B C D E F G
A B C D E F G H I
A B C D E F G H I J K
C D E F G H I J K
E F G H I J K
G H I J K
J I K
K
```

图 5－6　字符菱形(1)

图 5－6 的上半部分与图 5－5 相同，因此在程序 NO. 5－9 的基础上再添加打印下半部分的程序就行了。

程序见 No. 5－10。

```
10 REM No. 5－10, LING XING 1
20 CLS: A$ ="ABCDEFGHIJK": A=0: B=0
30 FOR I=1 TO 11 STEP 2
40 A=A+1: LOCATE 13－A,5+A
50 PRINT LEFT$ (A$ ,I)
60 NEXT
70 FOR I=9 TO 1 STEP －2
80 A=A－1: B=B+1: LOCATE 13－A,11+B
90 PRINT RIGHT$ (A$ ,I)
100 NEXT
110 LOCATE 0,22: END
```

程序中的 70～100 行为打印菱形下半部分的处理程序。最后一行的作用是把光标定位在屏幕的底部。

在 90 行使用的“RIGHT $”指令的作用是“从字符串右边取子字符串”。其语句格式为

RIGHT $(字符串,n)

功能是从字符串的右边取出“n”个字符组成子字符串。

程序 No. 5－10 使用了两个循环语句分别打印出菱形的上、下两部分，那是为了便于理解分段编写的。实际上，两个循环可以合并为一个，哪样程序将简洁的多。化简后的程序见 No. 5－11。

```
10 REM No. 5－11, LING XING 2
20 CLS: A$ ="ABCDEFGHIJK": A=0: B=0: C=6
30 FOR I=1 TO 11 STEP 2
40 A=A+1: B=B+1: C=C-1
50 LOCATE 13-A,5+A: PRINT LEFT$(A$,I)
60 IF I<11 THEN LOCATE 13-C,11+B: PRINT RIGHT$(A
   $,10-I)
70 NEXT
80 LOCATE 0,22: END
```

3. **从字串任意位置取子串** “MID $”指令可以实现在字符串中的任意位置取出任意个字符组成子字符串。其语句格式为

MID $(字符串,m,n)

语句功用是：从字符串的左边向右数第“m ”个字符位置开始，取“n”个字符组成子字符串。下面介绍该指令的应用。

例 5－4 编程画一个如图 5－7 所示的三角形。

实现程序见 No. 5－12。

```
10 REM No. 5－12, SAN JIAO XING 1
20 CLS: A$ ="ABCDEFGHIJ": A=0
30 FOR I=1 TO 10
40 A=A+1: B$ =MID$(A$,I,1)
50 LOCATE 5+A,8+A: PRINT B$
```

```
60 LOCATE 5+A,10-A:PRINT B$
70 IF I<10 THEN LOCATE 16,A*2-2: PRINT B$LOCATE
   16,A*2-1: PRINT B$
80 IF I=10 THEN LOCATE 16,A*2-2: PRINT B$
90 NEXT
100 LOCATE 0,22: END
```

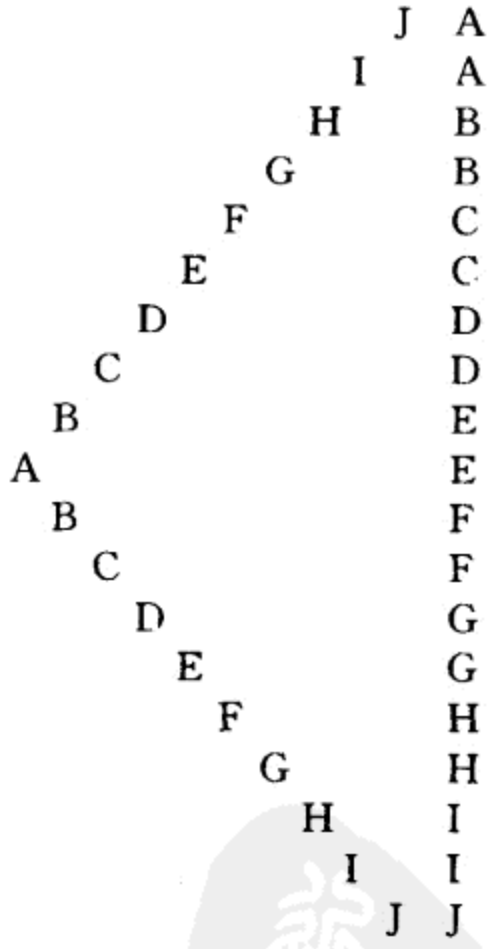


图 5－7　十字符三角形

程序 No. 5－12 中，50、60 两行语句，控制对称打印三角形的两个斜边；70 行控制打印直边。

三、打印三角形通用程序

程序 No. 5－12 只能打印以固定的字符串 A$ 中所含字

符的三角形，而且字符串中的字符个数限定于 10 个。能否编写一个通用的三角形打印程序，而且字符串的内容可以由键盘随机输入呢？

请看程序 No. 5－13。

```
10 REM No. 5－13, SAN JIAO XING 2
20 CLS: INPUT"INPUT A$";A$:
30 CLS: B=LEN(A$): A=0
40 FOR I=1 TO B
50 A=A+1: B$=MID$(A$,I,1)
60 LOCATE 5+A,B+A－2: PRINT B$
70 LOCATE 5+A,B－A:PRINT B$
80 IF I＜B THEN LOCATE B＋6, A ＊ 2 － 2: PRINT B
   $LOCATE B+6,A＊2－1: PRINT B$
90 IF I=B THEN LOCATE B+6,A＊2－2: PRINT B$
100 NEXT
110 LOCATE 0,22: PAUSE: GOTO 20
```

程序中使用了新指令“LEN”，它的功用是测试字符串的长度，其语句格式为

LEN(字符串)

程序 No. 5－13 中，20 行由用户输入字符串。字符串输入完毕，由 30 行测试输入字符串的长度赋于变量“B”。后面各行语句的作用与程序 No. 5－12 相同，语句形式不同的是：把与字符串长度有关的常数换成变量“B”，以使打印输出格式适应各种字符串长度。

运行程序 No. 5－13，屏幕显示

INPUT A$? ■

这时若输入字符串"12345"，回车后，清屏，打印的三角形如图 5－8 所示。

```
          5   1
        4     1
      3       2
    2         2
  1           3
    2         3
      3       4
        4     4
          5   5
```

图 5－8　五字符三角形

闪烁的光标等待用户输入下一个字符串。

按任意键后，清屏，显示

INPUT A$？ ■

输入"！@#$%^&*"，回车后，清屏，显示如图 5－9 所示的画面。

```
            *   !
          &     !
        ^       @
      %         @
    $           #
  #             #
@               $
!               $
@               %
  #             %
    $           ^
      %       ^
        ^   &
          &   &
            *   *
```

图 5－9　八字符三角形

该程序可一直运行下去，但要注意，由于屏幕的垂直分辨率为 24 行，所以输入的字符串长度最多为 11 个字符，否则将因输出超界而出错。因此，建议用户添加 35 行语句进行拦截。

35 IF B>11 THEN 20

添加 35 行后，当输入的字符串长度大于 11 时，将放弃当前的输入，要求用户重新输入字符串。

四、妙用字符函数

1. 菱形程序的优化

打印菱形的程序 No. 5－11 还不够精练，现在再介绍更优化的程序。见 No. 5－14。

```
10 REM No. 5－14, LING XING 3
20 CLS: A$ ="ABCDEFGHIJKLM": X=10: Y=0
30 FOR K=－4 TO 4: Y=Y+1
40 IF K<=0 THEN X=X－1: GOTO 60
50 X=X+1
60 LOCATE X,Y: PRINT MID$(A$,5－ABS(K),9－2 * ABS
   (K)): NEXT
70 LOCATE 0,22: END
```

运行程序，将打印出如图 5－10 所示的菱形。

```
        A
      B C D
    C D E F G
  D E F G H I J
E F G H I J K L M
  D E F G H I J
    C D E F G
      B C D
        A
```

图 5－10　字符菱形

程序 No. 5－14 输出的字符菱形特点是上下完全对称。

输出字符的对称和程序的优化，是取绝对值函数“ABS”要出的妙招。

“ABS”指令的语句格式为

ABS(X)

功能是对“X”取绝对值，即去掉“X”的符号。

欲弄清“ABS”如何要出妙招，须对程序 No. 5－14 运行中各变量的取值进行分析。

程序 No. 5－14 中，在“ABS”函数的作用下，循环变量“K”、字符显示坐标“X、Y”、“MID＄ ”语句中取字符的位置参量“m”和取字符数量参数“n”的对应关系如表 5－1 所示。

表 5－1　程序 No. 5－14 中有关变量、参数取值对照表

变量参量	取　　值								
K	－4	－3	－2	－1	0	1	2	3	4
X	9	8	7	6	5	6	7	8	9
Y	1	2	3	4	5	6	7	8	9
m	1	2	3	4	5	4	3	2	1
n	1	3	5	7	9	7	5	3	1

2. 菱形优化程序的变形

对程序 No. 5－14 稍加变形，即可打印出形态各异的图形，现仅列几例。

为便于观察，我们把程序 No. 5－14 中的字符串变量值由字母改为同一字符“＊”。

(1) 楔形

把 60 行稍微变动，就得到程序 No. 5－15。

```
10 REM No. 5－15, XIE XING
20 CLS: A$＝"＊＊＊＊＊＊＊": X＝10: Y＝0
30 FOR K＝－4 TO 4: Y＝Y＋1
40 IF K＜＝0 THEN X＝X－1: GOTO 60
50 X＝X＋1
60 LOCATE X,Y: PRINT MID$(A$,5－ABS(K),9－ABS
   (K)): NEXT
70 LOCATE 0,22: END
```

运行程序，将打印出图 5－11 所示的楔形。

```
        ＊＊＊＊＊
      ＊＊＊＊＊＊＊
    ＊＊＊＊＊＊＊＊＊
  ＊＊＊＊＊＊＊＊＊
＊＊＊＊＊＊＊＊＊＊
  ＊＊＊＊＊＊＊＊＊
    ＊＊＊＊＊＊＊
      ＊＊＊＊＊
        ＊＊＊
```

图 5－11　楔形

(2)三角形

把程序 No. 5－15 的 60 行改为

```
60 LOCATE X,Y: PRINT MID$(A$,5－ABS(K),
  5－ABS(K)): NEXT
```

运行程序，将打印出如图 5－12 所示的三角形。

图 5－12　三角形

(3)箭头

请见程序 No. 5－16

```
10 REM No. 5－16, JIAN TOU
20 CLS: A$ = "* * * * * * *": X = 10: Y = 0
30 FOR K = －4 TO 4: Y = Y + 1
35 IF K = 0 THEN X = X － 1: LOCATE X, Y: PRINT A$: GO-
   TO 70
40 IF K <= 0 THEN X = X － 1: GOTO 60
50 X = X + 1
60 LOCATE X, Y: PRINT MID$(A$, 5 － ABS(K), 1)
70 NEXT
80 LOCATE 0, 22: END
```

运行程序,打印结果如图 5－13 所示。

图 5－13　箭头

(4)锚形

程序见 No. 5－17。

```
10 REM No. 5－17, MAO XING
20 CLS: A$ =" * * * * *": X=10: Y=0
30 FOR K=－4 TO 4: Y=Y+1
35 IF K=0 THEN X=X－1: LOCATE X,Y: PRINT A$;A$;
   A$: GOTO 70
40 IF K<=0 THEN X=X－1: GOTO 60
50 X=X+1
60 LOCATE X,Y: PRINT MID$(A$,5－ABS(K),9－ABS
   (K))
70 NEXT
80 LOCATE 0,22. END
```

运行程序,打印结果如图 5－14 所示。

```
        * * * * *
      * * * *
    * * *
  * *
* * * * * * * * * * * * * * * * *
  * *
    * * *
      * * * *
        * * * * *
```

图 5－14　锚形

五、数字图形的困惑

例 5－5　编写打印数字菱形的通用程序,要求从外向内的每一层用同类数字围成,外层的数字最大,中间是“0”。

编程如 No. 5－18。

```
10 REM No. 5－18, SHU ZI LING XING
20 CLS: INPUT"INPUT 2－6=";N: CLS
30 FOR I=0 TO 2*N
40 K=K+1: LOCATE 7+2*ABS(N－I)－N,K
50 FOR J=N TO ABS(N－I) STEP －1
60 PRINT J;: NEXT
70 IF K=1 OR K=2*N+1 THEN 100
80 FOR P=J+2 TO N
90 PRINT P;: NEXT
100 NEXT
```

运行程序，键入“2”则打印如图 5－15 所示的图形。

```
      2
    2 1 2
  2 1 0 1 2
    2 1 2
      2
```

图 5－15　2 阶数字菱形

若键入“5”则打印如图 5－16 所示的图形。

由于屏幕显示分辨率所限，程序 No. 5－18 最大可打印 6 阶数字菱形。

```
        5
      5 4 5
    5 4 3 4 5
  5 4 3 2 3 4 5
5 4 3 2 1 2 3 4 5
  5 4 3 2 1 2 3 4 5
    5 4 4 2 3 4 5
      5 4 3 4 5
        5 4 5
          5
```

图 5－16　5 阶数字菱形

第四节　本章小结

本章共讨论了 10 条指令，现分别总结如下。

1. **随机数的产生(RND)**

格式：RND(X)

类型：函数

简写：RN.

功用：产生一个均匀分布的随机数。

随机数是事先无法预计的数，它由电脑根据语句中给定的参数(X)，通过特定的算法产生一个小于 X 的数字。

注释：RND 取自英文“RaNDom”，是“随机”的意思。

语句中的“X”取值范围为 1～32767 之间的任意整数。

示例：一位数加法练习程序见 No. 5－19。

```
10 REM No. 5－19 Jia fa lian xi
20 CLS : A$ ="                    ": GOTO 40
30 LOCATE 2,3: PRINT "Di";N;" ti, De fen=";F
40 A=RND(9)+1 : B=RND(9)+1 : N=N+1
50 LOCATE 6,8 : PRINT A"+"B"=" ; : INPUT "",C
60 IF C=A+B THEN B$ ="GOOD" : F=F+5 : GOTO 80
70 B$ ="WRONG"
80 LOCATE 7,10 : PRINT B$ ;
90 PAUSE : LOCATE 6,8 : PRINT A$ : LOCATE 7,10 :
   PRINT A$ : GOTO 30
```

运行程序，清屏，显示练习题如：

　　6＋1＝■

闪烁的光标等待输入答案，键入“7”回车后，显示

6+1=■

GOOD■

按任意键后，清屏，在上一行显示运算了第几题，得分是多少，在第二行显示下一道练习题，如

Di 0 ti，　　De fen= 0

3+2=■

练习可一直进行下去。

程序 No. 5－19 稍加改造，就可变为减法、乘法、除法和多位数练习程序。

2. **屏幕染色(COLOR)**

格式：COLOR<X，Y，n>

类型：语句

简写：COL.

功用：对屏幕上某个位置的背景图案染色。

注释：COLOR 是颜色的意思。

在 COLOR 语句的作用下，可对屏幕任意位置的背景图案染色。染色位置由“X、Y”决定。“X ”为屏幕水平方向的位置坐标，取值范围为 0～27；“Y”为垂直方向的坐标，取值范围为 0～23；“n”为选定的配色代码，取值范围为 0～3。“X、Y、n”均可以取常数、变量或表达式。

3. **调色板的选择(CGSET)**

格式：CGSET<m，n>

类型：语句或命令

简写：CG.

功用：规定背景和卡通显示页面的调色板。

注释：CGSET 取自“Color Grade SET”，为“设定颜色等级组”的意思。

m——表示背景调色板的编号。取值为0、1，它可以取变量或表达式的形式。

n——表示卡通调色板的编号。取值为0～2，也可以取变量或表达式的形式。

说明：

(1) 一般在程序的开始语句中使用CGSET语句，以决定背景卡通页的颜色组合。

(2) 当背景或卡通调色板中的颜色组合不符合要求时，要先使用CGSET语句选定调色板，再使用PALETB(S)语句设定配色号码中的颜色组合。

(3) 用户不设定调色板的编号时(即不使用CGSET)，系统的默认值为CGSET 1,1。

(4) COLOR语句中使用的配色号码即为由CGSET语句选定调色板中的编号。

(5) CGSET语句允许省略其中的一个参数，如

```
CGSET 1
```

表示省略了卡通调色板的设定参数。但在省略背景参数时，两参数间的逗号不能省略。如

```
CGSET ,1
```

4. 设定背景配色代码(PALETB)

格式：PALETB n,C1,C2,C3,C4

类型：语句

简写：PAL.B

功用：改变背景调色板中某配色代码的颜色组合。

使用PALETB语句可将背景任一调色板中四个配色代码中的颜色组合，进行任意设定(调色板号由CGSET语句选定)。

注释:PALETB 取自于“PALETte Background”,是对背景配色的意思。

n——为欲改变的配色代码,取值 0～3。

C1——为指定底背景颜色的颜色代码,n＝0 时有效。

C2～C4——为指定配色代码中的三种颜色代码。

5. **设定卡通配色代码(PALETS)**

格式:PALETS n,C1,C2,C3,C4

类型:语句

简写:PAL. S

功用:改变卡通调色板中某配色代码的颜色组合。

使用 PALETS 语句可将卡通任一调色板中四个配色代码中的颜色组合,进行任意设定(调色板号由 CGSET 语句选定)。

注释:PALETS 取自于“PALETte Sprite”,是对卡通配色的意思。

各参数的含义同 PALETB。

6. **从字串左边取子串(LEFT＄)**

格式:LEFT＄(字符串,n)

类型:函数

简写:LEF.

功用:从字符串的左边开始,向右取出“n ”个字符形成一个子字符串。

注释:LEFT＄是从左边取字符的意思。

n 为所取的字符个数,取值范围为 0～31,它可以取常数、变量和表达式。

字符串的字符个数不能多于 31 个。

7. **从字串右边取子串(RIGHT $)**

格式:RIGHT $(字符串,n)

类型:函数

简写:RI.

功用:从字符串的右边开始,取出"n "个字符形成一个子字符串。

注释:RIGHT $ 是从右边取字符的意思。

n 为所取的字符个数,取值范围为 0～31,它可以取常数、变量和表达式。

字符串的字符个数不能多于 31 个。

8. **从字串的任意位置取子串(MID $)**

格式:MID $(字符串,n,m)

类型:函数

简写:MI.

功用:从字符串中的"m"位置开始,取出"n"个字符形成一个子字符串。

注释:MIDdle 表示中间,MID $ 是从中间取字符的意思。

m——表示取字符的位置,要从字符串的左边向右数。其取值范围为 0～31,但不能大于字符串本身的长度。它可以是常数、变量或表达式。

n——表示从字符串的第 m 位向右取的字符个数。其取值范围为 0～31,但不能大于字符串的第 m 位后所剩字符的长度。它可以取常数、变量和表达式。

字符串的字符个数不能多于 31 个。

9. **测试字符串的长度(LEN)**

格式:LEN(字符串)

类型:函数

简写:LE.

功用:计算出字符串的字符个数。

使用 LEN 语句可以方便地求出字符串的长度。

注释:LEN 源于"LENgth",是长度的意思。

字符串可以是字符变量或表达式。当直接测试字符串的长度时,整个字符串必须用引号括起来。如

```
LEN(A$)
LEN("ABCDEFGHIJK")
```

当字符串以表达式的形式测试时,如

```
LEN(A$+B$)
```

A$和 B$的字符数之和不能大于 31 个。

10. **求绝对值(ABS)**

格式:ABS(X)

类型:函数

简写:AB.

功用:对 X 的数值取绝对值。

注释:ABS 是"ABSolute "(绝对)一词的缩写。

X 可以是－32768～32767 之间的任意数值,也可以是数值变量或算术表达式。

示例:

```
PRINT ABS(－122)
PRINT ABS(18/2－25)
```

执行后,将分别输出

```
122
 16
```

表示已把数据的符号去掉。

第六章
惊人的运算精度

我们知道，F BASIC 的数据容许范围为 －32768～32767，也就是说，参与运算的数据和结果都不能超出以上的数据范围。因此，有人断言：F BASIC 语言除了能编制一些简单的游戏外，别无长处。至于计算功能，有等于无，不堪大用。

那么，F BASIC 语言真的如此不济吗？

本章将通过巧妙地编程，以达到大型计算机的运算精度，来展现 F BASIC 的风采。

本章还结合运算程序的编写，介绍：

串、数转换指令	VAL
数、串转换指令	STR $
变量清零指令	CLEAR

第一节　运算方法的引入

如何打破 F BASIC 取值范围的限制，象浮点 BASIC 那样，方便地进行较大数据的各种运算，甚至优于浮点 BASIC 而进行各种高精度的计算是本章讨论的重点和目的。怎样达到这一目的呢？我们通过考察一个简单的加法运算实例引出解决这一问题的途径。

例 6－1 求 43725 加 44132 的和。

在笔算中，一般是列竖式计算。即：

$$\begin{array}{r} 43725 \\ +44132 \\ \hline 87857 \end{array}$$

计算结果是87857。

归纳起来，计算步骤共三步：

1. 写出两个加数，各位对齐。

2. 对应位分别相加，结果写在对应位的下方。

3. 把每位相加的结果，按高位在前、低位在后的顺序组成一个新的数字。

显然，上例的计算在F BASIC中是不能进行的(因两加数与运算和均超出数据容许范围)。但是，若把两加数的每一位分别装入数组A、B的各下标变量中，然后将A、B数组中对应下标变量的内容相加，相加之和存入数组C中，就可在F BASIC系统中进行计算。具体计算步骤如下：

1. 定义A(5)、B(5)、C(5)三个数组，把两加数分别装入数组A、B中，有

A(5)=4　A(4)=3　A(3)=7　A(2)=2　A(1)=5

B(5)=4　B(4)=4　B(3)=1　B(2)=3　B(1)=2

2. 数组A、B各变量内容对应相加，相加之和存入数组C中。有

C(5)=A(5)+B(5)=4+4=8

C(4)=A(4)+B(4)=3+4=7

C(3)=A(3)+B(3)=7+1=8

C(2)=A(2)+B(2)=2+3=5

C(1)=A(1)+B(1)=5+2=7

3. 将数组C各元素内容按高下标在前、低下标在后的顺序输出，即得结果

C(5)、C(4)、C(3)、C(2)、C(1)＝87857

比较两种运算过程，运算顺序一致，运算结果相同，证明此方法是可行的。

由于每个下标变量装入一位数字，下标变量的最大取值为255，所以引入数组参加运算，可以处理250位以内数字的运算。推而广之，若一个下标变量中装入两位、三位、四位数字，或者采用双下标变量，能够处理的数据位数可以扩展到千位以上。这样，即便是天文数字也可以任意处理了。

下面将深入浅出、由简到繁，分别介绍任意位数的加法、减法、乘法、除法运算程序的编写技巧。

第二节　基本子程序

为简化程序的编写过程，我们先熟悉几个带有工具性质的子程序。

一、把数据装入数组

F BASIC要求输入的数字小于32767，输入大于32767的数则显示错误，拒绝输入。但对输入的字符串，则可以接受31个字符。因此，把数字当做字符串输入，最高可达31位，这在一般的数学运算中，已完全可以满足要求。

1. **数据装入数组子程序**　把数字装入数组的方法有两种：一种方法是先告诉电脑要输入数据的位数，然后定义数组下标，利用循环语句把数据装入数组；另一种方法是把数据作为字符串输入后，由程序处理自动把数据装入数组。显然第二种方法较方便，但第一种方法的优点是可以装入大于31位的数据。我们仅讨论第二种方法，第一种方法留给读者练习。

把数据装入数组的程序见 No. 6－1。

```
5 REM No. 6－1, SUB 1
10 INPUT"SHU JU 1=";A$: A=LEN(A$)
15 INPUT"SHU JU 2=";B$: B=LEN(B$)
20 DIM C(A),D(B)
25 FOR I=1 TO A
30 C$=MID$(A$,I,1)
35 C(A－I+1)=VAL(C$): NEXT
40 FOR J=1 TO B
45 D$=MID$(B$,J,1)
50 D(B－J+1)=VAL(D$):NEXT
```

程序说明：

(1)10、15 行输入数据 1、2，并测量数据 1 的字符长度存入变量 A；数据 2 的字符长度存入变量 B。

(2)20 行定义 C(A)、D(B)数组。

(3)25～35 行把数据 1 装入 C(A)数组。

(4)40～50 行把数据 2 装入 D(B)数组。

2. **字符串向数值的转换** 程序 No. 6－1 中使用了一条新指令“VAL”，它的语句格式为

VAL(字符串)

它的功用是将字符串形数据或十六进制数据转换成十进制数据。

程序 No. 6－1 的 35 行就是把从字符串 A$ 中取出的子串 C$(数据 1 中的某一位)转换为数据，逐一赋值给数组 C 的下标变量，即把数据 1 的各位装入数组 C。50 行则是把数据 2 的各位装入数组 D。

二、加法子程序

1. 一般加法子程序

考虑到加法中的进位，求和运算子程序如 No. 6－2。

```
  5 REM No. 6－2, SUB 2
100 DIM E(A+1): X=1
105 FOR I=1 TO A
110 E(X)=C(I)+D(I)
115 IF E(X)>9 THEN E(X)=E(X)－10: E(X+1)=E(X+1)
    +1
120 X=X+1: NEXT
```

2. 通用加法子程序

实际上，可将两数相加之和存入两加数的任一数组中，从而省去一个数组，以减少内存占用。这只要把 C(A)数组中的下标调整为 A＋1 以存放进位，即可达到目的。程序见 No. 6－3。

```
5 REM No. 6－3, SUB 3
20 X=1: DIM C(A+1),D(B)
100 FOR I=1 TO A: C(X)=C(X)+D(I)
105 IF C(X)>9 THEN C(X)=C(X)－10: C(X+1)=C(X+1)
    +1
110 X=X+1: NEXT
```

三、通用数据装入数组子程序

读者可能已经发现，当数据 2 的位数多于数据 1 时，程序 No. 6－3 将因循环次数不足而丢失数据，因此必须在数据装入数组时，保证位数多的数据装入数组 C。改进后的数据装入数组通用子程序见 No. 6－4。

```
5 REM No. 6－4，SUB 4
10 INPUT "SHU JU 1="；A$：INPUT "SHU JU 2="；B$
15 A=LEN(A$)：B=LEN(B$)：IF A<B THEN SWAP A$，
   B$：GOTO 15
20 DIM C(A+1)，D(B)：X=1
25 FOR I=1 TO A：C(A－I+1)=VAL(MID$(A$，I，1))：
   NEXT
30 FOR J=1 TO B：D(B－J+1)=VAL(MID$(B$，J，1))：
   NEXT
```

四、运算结果的输出

运算结束后，结果存于数组 C 中，只要置一循环语句，依次从高下标到低下标取出各下标变量的内容排列成串，即得到输出结果。结果输出子程序见 No. 6－5。

```
5 REM No. 6－5，SUB 5
200 IF C(A+1)<>0 THEN A=A+1
205 FOR I=A TO 1 STEP －1：PRINT C(I)；：NEXT
```

五、结束处理

结束处理一般比较简单，若继续计算则使用 CLEAR 语句使变量清零，重新运行程序；若结束运算则进行停机操作。结束处理程序见 No. 6－6。

```
5 REM No. 6－6，SUB 6
300 PRINT：PRINT：JI XU MA？(Y/N)"；Y$=INKEY$(0)
310 IF Y$<>"N" THEN CLEAR：RUN
315 CLS：LOCATE 10，10：PRINT "GOODBYE!!!"：END
```

程序 No. 6－6 中使用了“CLEAR”指令，该指令有两个使用格式、两种功用。第一个功用是将内存中的所有数值变量

清零，将所有字符串变量清空，取消已定义的数组；第二个功用是规定用户使用的内存空间范围。

第三节　高精度加法运算

一、整数加法

把第二节介绍的子程序 No. 6－3～No. 6－6 四个子程序合并就组成了整数加法运算程序。见程序 No. 6－7。

```
1 REM No. 6－7, Zhengshu Jiafa
5 CLS: INPUT"SHU JU 1=";A$: INPUT"SHU JU 2=";B$
10 A=LEN(A$): B=LEN(B$): IF A<B THEN SWAP A$,
   B$: GOTO 10
15 DIM C(A+1),D(B): X=1
20 FOR I=1 TO A:C(A－I+1)=VAL(MID$(A$,I,1)):
   NEXT
25 FOR J=1 TO B:D(B－J+1)=VAL(MID$(B$,J,1)):
   NEXT
30 FOR I=1 TO A: C(X)=C(X)+D(I)
35 IF C(X)>9 THEN C(X)=C(X)－10: C(X+1)=C(X+1)+
   1
40 X=X+1: NEXT
45 IF C(A+1)<>0 THEN A=A+1
50 FOR I=A TO 1 STEP －1: PRINT C(I);: NEXT
55 PRINT: PRINT: JI XU MA? (Y/N)": Y$=INKEY$(0)
60 IF Y$<>"N" THEN CLEAR: RUN
65 CLS: LOCATE 10,10: PRINT "GOODBYE!!!": END
```

二、小数加法

同整数相比，小数多了小数点和小数部分。因此，一个小

数必须把整数、小数两部分分别装入两个数组中。这样，两个数据就要使用四个数组。计算时：整数、小数分别对应相加；相加后要判断小数的十分位是否有进位，若有进位则把进位加在整数的个位；最后把两部分连在一起，中间添上小数点即得计算结果。

1. **小数装入数组子程序** 关于小数加法的运算，与程序 No. 6－7 大同小异，仅是小数装入数组子程序稍有不同，现列于程序 No. 6－8。

```
5 REM No. 6－8, SUB 7
10 INPUT"SHU JU 1=";A$: A=LEN(A$)
15 FOR I=1 TO A: C$=MID$(A$,I,1)
20 IF C$="." THEN P=I: GOTO30
25 NEXT
30 Z=P－1: W=A－P: DIM Z1(Z),W1(W)
35 FOR I=1 TO Z:Z1(Z－I+1)=VAL(MID$(A$,I,1)):
   NEXT
40 FOR J=P+1 TO A: W1(J－P)=VAL(MID$(A$,J,1)):
   NEXT
45 INPUT"SHU JU 2=";B$: B=LEN(B$)
50 FOR I=1 TO B: D$=MID$(B$,I,1)
55 IF D$="." THEN Q=I: GOTO 65
60 NEXT
65 Y=Q－1: R=B－Q: DIM Z2(Y),W2(R)
70 FOR I=1 TO Y:Z2(Y－I+1)=VAL(MID$(B$,I,1)):
   NEXT
75 FOR J=Q+1TO B:W2(J－Q)=VAL(MID$(B$,J,1)):
   NEXT
```

运行程序 No. 6－8，可将两小数的整数部分分别装入数组 Z1、Z2；小数部分分别装入数组 W1、W2。

有了子程序 No. 6－8，再两次调用求和运算子程序和运算结果输出程序，并配以结束处理子程序，即可得出小数求和运算程序。但要注意，在进行小数部分求和时要从左至右对应位相加；在进行整数部分求和时，要从右至左对应位相加；求和运算时要注意循环变量终值的设置；编程中还要考虑小数部分向整数部分个位的进位和小数点的打印。该程序编写较容易，留给读者练习。

2. **通用加法运算程序**　其实，使用一个数组也可以编写小数加法运算程序。

下面要介绍程序的编写思路是：把参与运算的两个数据作为字符串处理，仅在相加时才把其变为数据；在运算前，先对两数据进行格式化，使整数、小数两部分位数相同，不足的部分用空格补齐。

另外，该程序还进行了笔算过程的模拟，列出竖式，演示进位操作，整个计算过程直观逼真。程序见 No. 6－9。

```
1 REM No. 6－9, Tongyong Jiafa
5 CLS: CLEAR: DIM R$(40): INPUT"X=";X$
10 INPUT"Y=";Y$: X1$=X$: Y1$=Y$:
   U$="                ": N1=1: N2=1: ML=MR
15 N1=N1+1: X2$=MID$(X$,N1,1): IF X2$="."
   THEN N1=N1－2: GOTO 25
20 GOTO 15
25 N2=N2+1: Y2$=MID$(Y$,N2,1): IF Y2$="."
   THEN N2=N2－2: GOTO 35
30 GOTO 25
35 IF N2<N1 THEN 50
40 IF N2=N1 THEN ML=N1: GOTO 55
45 X$=LEFT$(U$,N2－N1)+X$: ML=N2: GOTO 55
```

```
50 Y$=LEFT$(U$,N1-N2)+Y$: ML=N1
55 N1=LEN(X$)-ML: N2=LEN(Y$)-ML
60 IF N2<N1 THEN75
65 IF N2=N1 THEN MR=N2: GOTO 80
70 X$=X$+LEFT$(U$,N2-N1): MR=N2: GOTO 80
75 Y$=Y$+LEFT$(U$,N1-N2): MR=N1
80 N=ML+MR: CLS
85 FORI=1 TO N: B$=MID$(X$,I,1): C$=MID$(Y$,
   I,1)
90 LOCATE 9+I,2: PRINT B$: LOCATE 9+I,4: PRINT C
   $: NEXT
95 LOCATE 6,4: PRINT "+": LOCATE 4,8: PRINT "---
   ----------"
100 C=0
105 FORI=N TO 1 STEP-1
110 A=VAL(MID$(X$,I,1)): B=VAL(MID$(Y$,I,1)):
    Z=A+B+C: C=0
115 IF Z>9 THEN C=1
120 R$(I)=RIGHT$(STR$(Z),1)
130 LOCATE 9+I,10: PRINT R$(I)
135 IF I=ML+3 THEN I=I-1: R$(I)=".": GOTO 130
140 IF C=1 THEN LOCATE 7+I,6:PRINT C
145 PAUSE 200: NEXT
150 IF C=1 THEN R$(0)="1": LOCATE 9,10: PRINT"1":
GOTO 160
155 R$(0)=""
160 PRINT: PRINT X1$"+"Y1$"=";: FOR I=0 TO N:
    PRINT R$(I);: NEXT
165 LOCATE10,20: PRINT "Ji xu ma? (Y/N)";:A$=INKEY
    $(0): IF A$<>"N" THEN 5
```

```
170 CLS: LOCATE 10,10: PRINT "GOODBYE!": END
```

程序说明：

(1)15～30 行确定两数小数点位置。

(2)35～75 行对两数字进行格式化处理，使两数格式统一。如 X$＝“325.849”、Y$＝“3.9587”，格式统一后，X$＝“325.849 ”、Y$＝“ 3.9587”(在 X$ 从最后添了一个空格，在 Y$ 的前面添了两个空格)。其中 45、50 两行格式化整数部分，70、75 两行格式化小数部分。格式化后，两个字符串的长度一致，小数点位置相同。

(3)85～95 行列竖式，模拟笔算方式。

(4)100～155 行运算结果，并演示进位过程。

(5)160 行输出结果。

(6) 当输入的数据为整数时，必须在整数后加小数点和一个“0”。如输入“35”时应敲入“35.0”。

运行程序，屏幕的左上角显示

X＝? ■　　键入 325.849↙

Y＝? ■　　键入 3.9587↙

清屏后，显示

```
      3 2 5. 8 4 9
+       3 . 9 5 8 7
        1. 1 1
-------------------
    3 2 9. 8 0 7 7
```

325.849＋3.9587＝329.8077

JI XU MA? (Y/N)■

键入除“N”以外的任意键，可继续运算。

键入“N”键则清屏，显示“GOOD BYE!!!”

停机。

该程序模仿笔算逼真，是一个较好的加法运算程序。

第四节　高精度减法运算

有了加法运算程序的编写技巧和经验，只要分析处理好减法运算与加法运算的几个不同点，将很容易编写出减法运算程序。

1. 考虑差的正、负，当被减数小于减数时差为负值。
2. 解决两数位数相同时，大小的判别问题。
3. 处理好被减数小于减数时的借位。
4. 处理好结果的输出，使显示的差符合一般记数习惯。

如

325.11－316＝9.11

若打印为“009.11”则显得别扭。

一、整数减法

下面结合程序 No.6－10 讨论减法程序的设计思路。

```
5 REM No.6－10, Zhengshu Jianfa
10 CLS :CLEAR :INPUT "Bei jian shu=";A$ : INPUT "Jian
   shu=";B$: PRINT A$"－"B$"=";
15 A=LEN(A$): B=LEN(B$): IF A<B THEN SWAP A$,
   B$: SWAP A,B: PRINT "－";
20 DIM C(A+1),D(B)
25 C(A+1)=1: X=1
30 FOR I=1 TO A: C(A－I+1)=VAL(MID$(A$,I,1)):
   NEXT
35 FOR J=1 TO B: D(B－J+1)=VAL(MID$(B$,J,1)):
   NEXT
```

```
40 FOR I=1 TO B
45 IF C(X)<D(I) THEN C(X)=C(X)+10: C(X+1)=C(X+
   1)-1
50 C(X)=C(X)-D(I): X=X+1: NEXT
55 FOR J=B+1 TO A:IF C(X)<0 THEN C(X)=C(X)+10: C
   (X+1)=C(X+1)-1
60 X=X+1: NEXT
65 IF C(A+1)=0 THEN SWAP A$,B$ :PRINT "-"; : GO-
   TO 25
70 IF C(A)=0 THEN A=A-1: GOTO 70
75 FOR I=A TO 1 STEP -1 :PRINT RIGHT$(STR$(C(I)),
   1);: NEXT
80 PRINT :PRINT "Ji xu ma? (Y/N)": Y$=INKEY$(0): IF
   Y$<>"N" THEN 10
85 CLS :LOCATE 10,10: PRINT "GOODBYE!": END
```

程序说明：

(1)10～35 行把数据装入数组。若减数位数多于被减数，则打印“-”号，并将 A$、B$ 的内容交换，保证位数多的数据放在 A$ 中，并装入数组 C。其中 25 行的 C(A+1)= 1 是给被减数超高位预置“1”，用于两数位数相同时，判别差的符号。

(2)40～60 行为运算程序。其中 45 行用于当被减数小于减数时，被减数向高位“借一当十”后再进行相减。

(3)55～60 行的作用是：当两数相减结束后，判断被减数所剩高位是否为负数。若是负数则向高一位“借一当十”，使本位大于或等于零。高一位处理完毕后再处理高二位……，直到所有剩余高位都处理完毕。如

1000－325

之后，千位为“－1”，则通过 55 行千位向万位借一，使千位变为 9；这时万位又变为“－1”，则万位再向十万位借一，万位变为 9，十万位变为 0。

(4)当被减数、减数位数相同时，15 行不能判断两数的大小，只有在运算完毕，通过 65 行检查预置超高位的“1”是否被借走。若 C(A＋1)≠0 则说明被减数大，差为正数；若 C(A＋1)＝0 则说明被减数小，差为负数。要把 A$、B$ 内容对调，重新运算，输出负值的差。

(5)70～75 行是输出程序。70 行从差的高位向低位依次检查，以保证输出的差前面不带“0”，符合记数习惯。

二、小数减法

根据加、减运算的四个不同点，只要对通用加法程序稍加改造，即得通用减法程序。见 No. 6－11。

```
1 REM No. 6－11, Tongyong Jianfa
5 CLS : CLEAR : DIM R$(40): INPUT "X=";X$
10 INPUT "Y=";Y$: X1$=X$: Y1$=Y$: U$=" ":N1
   =1: N2=1: P=R=W=C
20 N1=N1+1: IF MID$(X$,N1,1)="." THEN 30
25 GOTO 20
30 N2=N2+1: IF MID$(Y$,N2,1)="." THEN 40
35 GOTO 30
40 IF N2<N1 THEN 55
45 IF N2=N1 THEN P=N1: GOTO 60
50 X$=LEFT$(U$,N2－N1)+X$: P=N2: GOTO 60
55 Y$=LEFT$(U$,N1－N2)+Y$: P=N1
60 N1=LEN(X$)－P : N2=LEN(Y$)－P
65 IF N2<N1 THEN 80
```

```
70 IF N2=N1 THEN R=N1: GOTO 85
75 X$=X$+LEFT$(U$,N2-N1): R=N2: GOTO 85
80 Y$=Y$+LEFT$(U$,N1-N2): R=N1
85 N=P+R: FOR I=1 TO N:IF VAL(MID$(X$,I,1))<
   VAL (MID$(Y$,I,1)) THEN SWAP X$,Y$: W=1:
   GOTO 95
90 NEXT
95 CLS : LOCATE 10,4 : PRINT X$: LOCATE 10 ,5: PRINT
   Y$
100 LOCATE 6,5: PRINT "-": LOCATE 4,7: PRINT "---
    ----------"
105 FOR I=N TO 1 STEP -1
110 A=VAL(MID$(X$,I,1)): B=VAL(MID$(Y$,I,1)):
    IF A<(B+C) THEN Z=A+10-B-C: C=1: GOTO 118
115 Z=A-B-C: C=0
118 R$(I)=RIGHT$(STR$(Z),1)
120 LOCATE9+I,8 : PRINT R$(I)
125 IF I=P+1 THEN I=I-1: R$(I)=".": GOTO 120
130 IF C=1 THEN LOCATE 8+I,6: PRINT "1"
135 PAUSE 200: NEXT
140 IF R$(N)="0" THEN N=N-1: GOTO 140
145 K=0 :PRINT :IF W=1 THEN PRINT X1$"-"Y1$"=
    -"; :GOTO 155
150 PRINT X1$"-"Y1$"=";
155 K=K+1:IF R$(K)="0" THEN 155
160 FOR I=K TO N: PRINT R$(I);: NEXT
165 LOCATE 10,20: PRINT "Ji xu ma?(Y/N)"; :A$=INKEY
    $(0): IF A$<>"N" THEN 5
170 CLS : LOCATE 10,10: PRINT "GOODBYE!": END
```

程序说明：

(1)105～135 行完成减法运算。

(2) 140～155 行对差整形，使输出的结果符合记数习惯，并打印算术横式。

(3) 输入整数时，要求在数尾加小数点和一个“0”。如输入“15”时要键入“15. 0”。

(4) 本程序未加结束处理，若需要可自己添加。

三、数—串转换指令(STR $)

程序 No. 6－10 的 118 行使用了一条新指令“STR $”，其语句格式为

STR $ (X)

功用是把十进制数据“X”转换为字符串。

由于 F BASIC 的输出指令“PRINT”输出数据时，要带符号，若输出的数据是负数则数字前面打印“－”号，若为正数则数字前面空一格。因此，为避免列式演示中差的各位之间距离太大，故把各位数字转换为字符输出，以保证输出的数据紧凑。

第五节　高精度加减混合运算

有了小数的加、减通用运算程序 No. 6－9、No. 6－11，我们可以着手设计代数加、减混合运算的处理程序。

可以看出程序 No. 6－9、No. 6－11 只适用正数的运算，这在小学算术运算中可以满足要求。然而在代数运算中，可能遇到负数加正数、负数减正数等多种情况。因此，当符号不同时要用绝对值大的数减绝对值小的数，然后再根据绝对值大的数的符号决定差的符号。运算结果既要考虑绝对值的大小

还要考虑差的正负。另外，在实际运算中，往往不仅是两个数相加、减，可能要对一组数据进行连续的加、减运算，这就要把上一次的运算结果转化为下一次运算的数据，再与新输入的数据进行运算。因而，在程序的设计中，要周密考虑、全面安排，以保证程序的正确性和通用性。

本程序不仅可以作为中学生学习代数的辅助教学软件，还可以作为会计、统计等专业人员的计算工具。

1. 通用加、减演示运算程序

通用加、减演示运算程序见 No. 6－12。

```
1 REM No. 6－12, Tongyong Jia－Jian fa
5 CLS : CLEAR : DIM R$(40) : INPUT "X=";X$ : X1$=X
  $ : X1=LEN(X$) : U$=" "
10 INPUT "Y=";Y$ : Y1$=Y$ : Y1=LEN(Y$) : N1=0:
  N2=0 : C=0 : P=0 : R=0 : U=0 : V=0 : W=0 : X1$=
  X$ : X1=LEN(X$)
15 IF LEFT$(X$,1)="－"THEN U=1 : X1=X1－1: X$=
  RIGHT$(X$,X1)
20 IF LEFT$(Y$,1)="－" THEN V=1 : Y1=Y1－1: Y$=
  RIGHT$(Y$,Y1)
25 N1=N1+1: IF N1=X1+1 THEN X$=X$+".0": GOTO
  35
30 IF MID$(X$,N1,1)<>"." THEN 25
35 N2=N2+1: IF N2=Y1+1 THEN Y$=Y$+".0": GOTO
  45
40 IF MID$(Y$,N2,1)<>"."THEN 35
45 IF N2<N1 THEN 60
50 IF N2=N1 THEN P=N1 : GOTO 65
55 X$=LEFT$(U$,N2－N1)+X$ : P=N2 : GOTO 65
60 Y$=LEFT$(U$,N1－N2)+Y$ : P=N1
```

```
65 N1=LEN(X$)-P : N2=LEN(Y$)-P
70 IF N2<N1 THEN 85
75 IF N1=N2 THEN R=N1 : GOTO 90
80 X$=X$+LEFT$(U$,N2-N1) : R=N2 : GOTO 90
85 Y$=Y$+LEFT$(U$,N1-N2) : R=N1
90 N=P+R :CLS :IF V=1 THEN PRINT X1$;Y1$;"=?":
   GOTO 100
95 PRINT X1$"+"Y1$"=?"
100 GOTO 350
105 LOCATE 10,4 : PRINT X$ : LOCATE 10,5 :PRINT Y$
110 LOCATE 4,7 : PRINT "----------------"
115 IF U=V THEN LOCATE 6,5 : PRINT "+" :GOSUB 200
120 IF U<>V THEN LOCATE 6,5 : PRINT "-" :GOSUB
    300
125 IF R$(N)="." THEN N=N-1 : GOTO 135
130 IF R$(N)="0" THEN N=N-1 : GOTO 125
135 K=-1 : X$="0" : PRINT : PRINT
140 K=K+1: IF R$(K)="0" OR R$(K)="" THEN 140
145 IF R$(K)="." THEN R$(K)="0."
150 IF (U=0 AND V=0)OR(U=1 AND V=0 AND W=1)
    THEN PRINT X1$;"+";Y1$;"=";
155 IF (U=1 AND V=1)OR(U=0 AND V=1 AND W=1)
    THEN X$="-" : PRINT X1$;Y1$"=-";
160 IF U=1 AND V=0 AND W=0 THEN X$="-" : PRINT
    X1$;"+";Y1$;"=-";
165 IF U=0 AND V=1 AND W=0 THEN PRINT X1$;Y1$"
    =";
170 FOR I=K TO N : PRINT R$(I); :X$=X$+R$(I) : R
    $(I)="" : NEXT
175 X=LEN(X$) : IF LEFT$(X$,1)="0" THEN X$=
```

```
    RIGHT$(X$,X-1)
180 LOCATE 0,20 : PRINT "Lian xu ji suan ma? (Y/N)"; : Q
    $=INKEY$(0) : IF Q$<>"N" THEN PRINT : PRINT
    "X= "X$ : GOTO 10
185 LOCATE 0,22:PRINT "Chong xin ji suan ma?(Y/N)" ; : Q
    $=INKEY$(0) : IF Q$<>"N" THEN 5
190 CLS : LOCATE 10,10 : PRINT "GOODBYE!" : END
200 FOR I=N TO 1 STEP -1
205 A=VAL(MID$(X$,I,1)) : B=VAL(MID$(Y$,I,1))
    : Z=A+B+C : C=0
210 IF Z>9 THEN C=1
215 R$(I)=RIGHT$(STR$(Z),1)
220 LOCATE 9+I,8 : PRINT R$(I)
225 IF I=P+1 THEN I=I-1 : R$(I)="." : GOTO 220
230 IF C=1 THEN LOCATE 8+I,6 : PRINT "1"
235 PAUSE 200 : NEXT
240 IF C=1 THEN R$(0)="1": LOCATE 9,8 : PRINT "1"
245 RETURN
300 FOR I=N TO 1 STEP -1
305 A=VAL(MID$(X$,I,1)) : B=VAL(MID$(Y$,I,1)):
    IF A<B+C THEN Z=A+10-B-C : C=1 : GOTO 315
310 Z=A-B-C : C=0
315 R$(I)=RIGHT$(STR$(Z),1)
320 LOCATE 9+I,8 : PRINT R$(I)
325 IF I=P+1 THEN I=I-1 : R$(I)="." : GOTO 320
330 IF C=1 THEN LOCATE 8+I,6 : PRINT "1"
335 PAUSE 200 : NEXT
```

```
340 RETURN
350 FOR I=1 TO N : AA=VAL(MID$(X$,I,1)) : BB=
    VAL(MID$(Y$,I,1)) : IF AA>BB THEN 105
355 IF AA<BB THEN SWAP X$,Y$ : W=1 : GOTO 105
360 NEXT : IF U=V THEN 105
365 PRINT "Jie guo='0'" : X$="0" : GOTO 180
```

2. 程序说明

(1) 25～35 行的作用是对输入的整数，自动补小数点和添加一个“0”，以方便格式化处理。这样，对输入数据没有任何特殊要求。

(2) 45～85 行完成两数的格式化。

(3) 90、95 两行打印问题的横式。

(4) 105～110 行列出运算竖式。

(5) 125～145 行对运算结果整形。其中 125 行对整数结果去小数点；130 行去除结果小数后的“0”；140 行去除结果整数部分前面的“0”；145 行对无整数部分结果的前面添加“0.”。125、145 行的作用就是使输出的结果符合人们的记数习惯。

(6) 200～245 行为加法运算子程序。

(7) 300～340 行为减法运算子程序。

(8) 350～365 行为比较|X|、|Y|大小的子程序。

3. 程序运行示例

运行程序，屏幕显示

X=？ ■　　键入－112.53↙

Y=？ ■　　键入 3264.2↙

清屏。显示

－112.53+3264.2=？

3264.2
－ 112.53
1.1

3151.67

－112.53＋3264.2＝3151.67

LIAN XU JI SUAN MA？（Y/N）（询问“连续计算吗”）

若连续计算，则键入除“N”键外的任意键，则显示

X＝3151.67（把上一次的运算结果转为当前运算的数据）

Y＝？ ■　　键入－6127.653↙

清屏。显示

3151.67－6127.653＝？

6127.653
－ 3151.67
1111

2975.983

3151.67－6127.653＝2975.983

LIAN XU JI SUAN MA？（Y/N）（询问“连续计算吗”）

若键入“N”则显示

CHONG XIN JI SUAN MA？（Y/N）（询问“重新计算吗”）

若重新计算则键入除“N”键外的任意键。

若结束运算则键入“N”键。

清屏。显示

GOODBYE!!!

停机。

4. **程序使用说明**

(1) 本程序可对任意整数、小数、带符号数进行 30 位以内的加、减运算。各种数字的输入可按正常方式操作，没有特殊要求。

(2) 由于要在屏幕上列竖式演示运算过程，受屏幕限制，输入数据的位数限于 23 位以下。若要进行更多位数据的计算，可删除列竖式语句。

(3) 程序清单是按演示运算给出的(最完全的程序)。若用于会计、统计计算，可删除列竖式语句 235、335 行中的延时语句，并将 180 行改为"GOTO 10"则可方便地完成任意个数据的连续加减。

第六节　高精度乘法运算

编写计算机的乘法运算程序并不难，但编写高精度乘法运算程序则不那么简单。我们知道，微机虽然已升档到 286、386、486……，其运算能力也越来越强，但对于大于 9 位的数据就采用了科学记数法，其运算精度并不高。因此，在 F BASIC 系统下进行高精度运算确实需要动一番脑筋。

编写高精度运算程序仍然离不开数组。仔细分析，乘法运算程序的数据装入数组、结果输出和结束处理与加、减法程序的区别不大，关键在运算部分。下面仅介绍整数高精度乘法运算程序的编写方法，小数高精度乘法运算程序将放在丛书之九《BASIC 程序集》中介绍。

1. **编程思路的引入**

首先看一个笔算三位数乘法的运算过程。如

$$987\times654=?$$

一般列竖式计算如下：

```
      9 8 7
    ×6 5 4
  ---------
      3 9 4 8
    4 9 3 5 0
  5 9 2 2 0 0
  ---------
  6 4 5 4 9 8
```

运算过程共分四步：

(1) 用乘数的个位“4”分别乘以被乘数的各位，得出乘积“3948”。

(2) 用乘数的十位“50”分别乘以被乘数的各位，得出十位的乘积“49350”

(3) 用乘数的百位“600”分别乘以被乘数的各位，得出百位的乘积“592200”。

(4) 对三个乘积求和，得出最终的乘积“645498”。

2. 编程思路

根据笔算过程的总结，可以确定我们的设计思路为：

(1) 定义一个装乘积的数组，把第一次运算的乘积依次装入数组中。

(2) 把高一位的乘积依次与前一次的乘积相加，所得之和再存入乘积数组的对应单元中。

(3) 乘数有多少位则进行多少次“(2)”的操作。

(4) 与笔算的不同处是：不是最后求和，而是进行一位求积运算，进行一次求和。

3. 程序清单

高精度整数乘法运算程序见 No. 6－13。

```
1 REM No. 6－13，Zhengshu Chufa
5 CLS ：CLEAR ：INPUT "Bei cheng shu B＝";B$ ：P＝LEN
  (B$)
10 INPUT "Cheng shu C＝";C$ ：Q＝LEN(C$)
15 G＝P＋Q ：DIM A(G),B(P),C(Q)
20 FOR I＝1 TO P ：B(P－I＋1)＝VAL(MID$(B$,I,1)) ：
   NEXT
25 FOR J＝1 TO Q ：C(Q－J＋1)＝VAL(MID$(C$,J,1)) ：
   NEXT
30 FOR K＝1 TO Q ：X＝K
35 FOR J＝1 TO P ：D＝B(J)＊C(K) ：A(X)＝A(X)＋D
40 IF A(X)＞9 THEN A(X＋1)＝A(X＋1)＋A(X)/10 ：A(X)＝
   A(X)－A(X)/10＊10
45 IF A(X＋1)＞9 THEN A(X＋2)＝A(X＋2)＋1 ：A(X＋1)＝
   A(X＋1)－A(X＋1)/10＊10
50 X＝X＋1 ：NEXT
55 NEXT ：PRINT ：PRINT B$;"X";C$;"＝";
60 IF A(G)＝0 THEN G＝G－1
65 FOR I＝G TO 1 STEP －1
70 PRINT RIGHT$(STR$(A(I)),1); ：NEXT
```

4. 程序说明

(1) 30～55 行是乘法运算程序。本程序采用两重循环，外层循环控制乘数各位，内层循环控制被乘数各位。通过外层循环，先取乘数个位，进入内循环后依次与被乘数相乘，相乘之积存入数组 A(G)中。

(2) 35～50 行完成一位乘数与被乘数各位相乘的操作。

(3) 40～45 行完成进位操作。由于两个单位数相乘之积 A(X)中最大的数为 81，因此进位最大为 8，而 A(X＋1)位由于与 A(X)的进位相加有可能大于 9，故还须考虑向 A(X＋

2)位的进位问题。

乘法程序的编程技巧关键在于进位处理，只要理解掌握了 40～45 行就掌握了乘法的编程技巧。

显然，程序 No. 6－13 可以进行乘积为 255 位的高精度乘法运算(因乘积数组的最大下标为 255)。若想进行更大数据的运算，可定义双下标乘积数组。

好了，有了程序 No. 6－14，我们可以做一些有趣的计算，下列计算结果有些即使在大型微机上也难以看到。

数学中把左右对称的数叫回文数，如：

3 6 3

2 6 7 6 2

1 9 9 1

等都叫回文数。自然数中有无限个回文数，你见过下面的回文数吗？

① 12345678987654321

② 121242363484363242121

③ 1210242036304840363024201 21

④ 123210245420369630246420123 21

⑤ 10203040506070809080706050403020 1

⑥ 1002003004005006007008009008007006005004003002001

不要以为这些数是杜撰的，请输入下列算式试一试。

11111111×11111111

11011011011×11011011011

11001100110011×11001100110011

11100011100011 1×11100011100011 1

10101010101010101×10101010101010101

100100100100100100100100 1×100100100100100100100100 1

第七节　高精度除法运算

编程思路：我们可以模仿人们笔算除法的过程，先求商的整数部分，把它打印出来；然后把余数扩大 10 倍（即笔算除法中的不够除补“0”），用余数去除以除数，求得小数点后第一位；再把余数扩大 10 倍，再用余数去除以除数，求得小数点后的第二位。如此，循环往复，直到所希望的精度为止。

一、简单整数的高精度除法

1. **编程思路**　以 X 表示被除数、Y 表示除数、N 表示运算精度（即小数点后的位数）；以 M 为商（M＝X/Y），安排一个循环语句，以 N 为循环终值，每计算一位即输出一位，循环结束即达到了运算精度。循环程序如下：

```
40 FOR I=1 TO N
45 X=(X-M*Y)*10
55 M=X/Y: PRINT M;
60 NEXT
```

程序中 45 行的 M 是小数点后上一位的商，N 为要求的精度，N 为多少，就算到多少位。

对于“0”不能作除数的问题，可安排一判断语句进行处理。

```
10 INPUT"X,Y,N=";X,Y,N
15 PRINT X"/"Y"=";
20 IF Y=0 THEN PRINT"ERROR! INPUT AGAIN!": GO-
   TO10
```

判断商的正负，可如下安排语句：

```
25 IF (X>0 AND Y<0) OR (X<0 AND Y>0)THEN PRINT"
```

--";

当处理完商的符号后，把被除数、除数均以正数处理。即

```
30 X=ABS(X): Y=ABS(Y)
```

然后，利用整数 BASIC 对商去尾取整的特点，可方便地求出商的整数部分 M，打印 M，并在 M 后打印小数点。

```
35 M=X/Y: PRINT M;".";
```

达到运算精度后，对最后一位小数进行四舍五入处理。

```
50 IF I=N THEN 60
60 M1=X/Y: IF M1>5 THEN M=M+1
65 PRINT M: END
```

至此，全部程序设计完成，汇总合成后的程序即得到完整的除法运算程序。

2. 程序清单

程序清单见 No. 6－15。

```
1 REM No. 6－15, Zhengshu Chufa
5 CLS : CLEAR
10 INPUT "X/Y.....N=";X,Y,N
15 PRINT X"/"Y"=";
20 IF Y=0 THEN PRINT "Error! Input again!" : GOTO 10
25 IF ( X>0 AND Y<0 ) OR ( X<0 AND Y>0 ) THEN
   PRINT "－";
30 X=ABS(X) : Y=ABS(Y)
35 M=X/Y : PRINT RIGHT$(STR$(M),1);".": : X=(X－
   M*Y)*10
40 FOR I=1 TO  N
45 M=X/Y : X=(X－M*Y)*10
50 IF I=N THEN 60
55 PRINT RIGHT$(STR$(M),1): : NEXT
```

```
60 M1=X/Y : IF M1>=5 THEN M=M+1
65 PRINT RIGHT$(STR$(M),1)) : END
```

若要连续运算，读者可在 65 行后加结束处理程序。

现在让我们检验一下程序 No. 6－15 的功能。

我们知道圆周率“π”的密率是

$$\pi=355/113$$

但它仍不是精确的“π”。试验证密率在小数点后多少位后开始循环。运行程序，键入

```
355,113,300
```

回车后，将打印

```
355/113=3.1415929203539823
8849557522123893805309734513‘
2743362831858407079646017699
1150442477876106194690265486
7256637168141592920353982300
8849557522123893805309734513
2743362831858407079646017699
1150442477876106194690265486
7256637168141592920353982300
8849557522123893805309734513
2743362831858407079646017699
11
OK
■
```

可见，圆周率的密率在小数点后第 112 位开始循环。

程序 N0. 6－15 只能进行－32768～32767 之间的任意整数的高精度除法运算。

二、任意整数的高精度除法

欲进行任意数的高精度除法，仍须借助于字符串变量和数组才能实现。

由于高精度除法的程序结构较复杂，受篇幅所限，不再讨论编程原理和进行程序说明。有兴趣的读者请参见《电脑游戏机 F BASIC 语言与编程技巧》一书。

任意整数的高精度除法运算程序见 No. 6－16。

```
1 REM No. 6－16, Zhengshu Chufa
2 CLS
5 CLEAR :U=0 :V=0 :T=0 :C=0 :D=0 :E=0 :G=0
10 INPUT "X/Y.....N=";X$,Y$,N :PRINT X$;"/";Y
   $;"="; :A=LEN(X$) :B=LEN(Y$) : I=1
15 XX=VAL(MID$(X$,I,1)) :IF XX>0 THEN 30
20 I=I+1 :IF I<=A THEN 15
25 PRINT "0" : END
30 I=1
35 YY=VAL(MID$(Y$,I,1)) :IF YY>0 THEN 45
40 I=I+1 :IF I<=B THEN 35 :PRINT "Error! Input again!" :
   GOTO 5
45 IF LEFT$(X$,1)="－" THEN A=A－1 : U=1 :X$=
   RIGHT$(X$,A)
50 IF LEFT$(Y$,1)="－" THEN B=B－1 : V=1 :Y$=
   RIGHT$(Y$,B)
55 IF U<>V THEN PRINT "－";
60 DIM A(B),B(B),C(B)
65 FOR I=1 TO B : B(I)=VAL(MID$(Y$,I,1)) : NEXT
130 A=LEN(X$)
135 IF A<B AND G=0 AND C=0 AND D=0 THEN PRINT "
```

```
    0.";:X$=X$+"0":A=A+1:C=1:G=1:D=1
140 IF A<B AND C=0 THEN PRINT ".";:X$=X$+"0":
    A=A+1:C=1:D=1
145 IF A<B AND D=0 THEN X$=X$+"0":A=A+1:D
    =1
150 IF A<B AND D=1 THEN PRINT "0";:X$=X$+"0":
    A=A+1:C=C+1:GOTO 150
155 X1$=LEFT$(X$,B):IF A=B THEN X2$="?":GO-
    TO 165
160 X2$=RIGHT$(X$,A-B)+"?"
165 I=1
170 X1=VAL(MID$(X1$,I,1)):Y1=VAL(MID$(Y$,I,
    1)):IF X1>Y1 THEN 185
175 IF X1<Y1 THEN 195
180 I=I+1:IF I<=B THEN 170
185 FOR I=1TO B:A(I)=VAL(MID$(X1$,I,1)):NEXT
190 A(0)=0:GOTO 240
195 IF A>B AND E=1 THEN PRINT "0";:GOTO 225
200 IF A=B AND G=0 AND C=0 AND D=0 THEN PRINT "
    0.";:X$=X$+"0":C=1:A=A+1
205 IF A=B AND E=1 AND C=0 THEN PRINT "0.";:X$=
    X$+"0":C=1:A=A+1
210 IF A=B AND C=0 THEN PRINT ".";:X$=X$+"0":
    A=A+1:C=1
215 IF A=B AND D=0 THEN X$=X$+"0":A=A+1
220 IF A=B AND D=1 THEN PRINT "0";:X$=X$+"0":
    A=A+1:C=C+1
225 A=LEN(X$):X1$=LEFT$(X$,B+1):IF A=B+1
    THEN X2$="?":GOTO 235
230 X2$=RIGHT$(X$,A-B-1)+"?"
```

```
235 FOR I=0TO B : A(I)=VAL(MID$(X1$,I+1,1)) :
    NEXT
240 X=B
245 I=B
250 IF A(0)*10+A(1)<B(1) THEN 275
255 IF A(X)<B(I) THEN A(X)=A(X)+10 : A(X-1)=A(X
    -1)-1
260 A(X)=A(X)-B(I) :X=X-1 :I=I-1 :IF I>0 THEN 250
265 IF X=0 THEN FOR J=1 TO B : C(J)=A(J) : NEXT
270 T=T+1 : GOTO 240
275 PRINT RIGHT$(STR$(T),1); : E=0 : T=0 : Z$="0"
280 FOR I=1 TO B :Z$=Z$+RIGHT$(STR$(C(I)),1) :
    NEXT
285 IF X2$="?"THEN X$=Z$ : GOTO 340
290 Z=LEN(Z$) : X2=LEN(X2$) :Z=Z-1 :Z$=RIGHT
    $(Z$,Z): X2=X2-1 : X2$=LEFT$(X2$,X2)
295 IF X2>0 THEN Z$=Z$+LEFT$(X2$,1) :Z=Z+1 :
    X2=X2-1
300 IF X2>0 THEN X2$=RIGHT$(X2$,X2)
305 IF LEFT$(Z$,1)="0" AND Z>1 THEN Z$=RIGHT
    $(Z$,Z-1) : Z=Z-1 : E=1 : GOTO 305
310 IF Z=1 AND Z$="0" AND X2=0 THEN PRINT "0"; :
    END
315 IF Z=1 AND Z$="0" AND X2>0 THEN PRINT "0"; :
    GOTO 295
320 IF X2=0 AND Z<B THEN PRINT"0"; :X$=Z$ :GOTO
    360
325 IF X2=0 AND Z>=B THEN X$=Z$ :GOTO 360
330 IF Z<B THEN PRINT "0"; : GOTO 295
335 X$=Z$+X2$
```

```
340 A=LEN(X$) :IF A=1 AND X$="0" THEN END
345 IF LEFT$(X$,1)="0" THEN X$=RIGHT$(X$,A-
    1) :GOTO 340
350 IF C>0 THEN C=C+1
355 IF C>N THEN END
360 D=0 : G=1 : GOTO 130
```

现在我们来验证一下回文数是否正确，运行程序，显示

X/Y.....N=？ ■

提示的意思是：请输入被除数、除数和商的小数点后精确的位数。输入

12345678987654321,111111111,1

回车后，将打印出结果

12345678987654321/111111111=111111111

第八节　本章小结

一、本章要点

本章重点介绍了任意数的高精度加、减运算和任意整数的高精度乘、除运算程序，受篇幅所限仅简单地讨论了编程思路，主要是给出源程序清单，供大家借鉴和使用。至于通用的乘法高精度运算程序、通用的除法高精度运算程序以及求幂、开方等高精度运算程序，由于需要考虑的问题较多、程序的结构较复杂，不利于初学者接受，没有涉及，有兴趣的朋友可继续阅读拙作“电脑游戏机《F BASIC 语言与编程技巧》”一书。

虽然本书仅讨论了简单的高精度运算程序，但其数百位乘积的计算、数十位回文数的验证、圆周率密率循环位数的求

得等运算，已令人对 F BASIC 刮目相看。特别是带有计算过程演示的通用加、减运算程序，既可充当中、小学生的数学家庭教师，又可作为会、统人员的计算工具，其实用价值是不言自明的。

谁说 F BASIC 计算能力有限？

本章的叙述就是有力的证明。

二、指令总结

本章共讨论了三条指令“VAL、CLEAR、STR $”。

1. 串—数转换指令(VAL)

格式：VAL(字符串)

类型：函数

简写：VA.

功用：

(1) 将十六进制数转换成相应的十进制数。

(2) 将字符串型的数据转换成十进制数。

就是说，VAL 函数具有去掉字符串引号、字符的功能。VAL 的返回值可以参加数值运算。

注释：VAL 源于“VALue”，是数值的意思。

(1) 当 VAL 函数用于数据的进制转换时，十六进制数必须在数值前面加“&”或“&H”，且用双引号括起来。如

“&32”

“&HAF00”

等。十六进制数的取值范围为 &H00 ～ &HFFFF 。当取值为 &H00～&H7FFF 时，返回的十进制数是 0～32767 ；当取值为 &H8000～&HFFFF 时，返回的十进制数是－32768～－1。

(2) 当 VAL 函数用于串型数据转换时，其返回值仅保留字符串前面的数码部分，字符部分则删除。下面把串变量取不同值时 VAL 函数返回的结果列于表 6－1 供参考。

表 6－1　串变量取不同值时 VAL 函数返回值对照表

串变量　A$	VAL(A$)
“1235”	1235
“－367”	－367
“45CDE”	45
“HELLO3”	0

2. **数一串转换指令(STR$)**

格式：STR$(X)

类型：函数

简写：STR.

功用：将十进制数转换成相应的字符串。

注释：STR$取自于“STRing”，为“串”的意思。

“X”的取值范围为－32768～＋32767。可以取常数、变量或表达式。STR$函数会将“X”的值转换成相应的串型数。

如果“X”为负值，则 STR$ 函数生成字符串的第一个字符为“－”；如果“X”为正值则 STR$ 函数生成字符串的第一个字符为空格。

示例：编程演示 STR$函数的功用。

```
10 CLS:
20 INPUT"A,B=";A,B
```

```
30 PRINT A"+"B"="A+B
40 PRINT "STR $ ("A")+STR $ ("B"=";STR $ (A)+STR
   $ (B)
50 GOTO 20
```

运行程序，清屏，显示

A，B=？ ■　　键入 125，135

125+135=260

STR $ (125)+STR $ (135)=125135

A，B=？ ■

可见，通过 STR $ 函数的处理，原来的十进制数 125、135 虽然外形没有变，但已发生了质的变化。这时它们已变为字符。因此，当相加时，不再按数字加法的规则进行而按字符串的规则相加了。

3. **变量清零指令(CLEAR)**

CLEAR 指令有两个格式、两种功用，下面分别介绍。

格式 1：CLEAR

类型：语句、命令

简写：CLE.

功用：将内存中的所有数值变量清零、所有字符变量清空、取消已定义的数组。

注释：CLEAR 为“清除”的意思。

格式 2：CLEAR＜数值＞

类型：语句、命令

简写：CLE. ＜数值＞

功用：规定用户使用的内存空间范围。

为了防止因用户程序太长而与 BASIC 系统内存空间相

冲突，使用 CLEAR 语句规定用户程序在内存中存放的起始位置。如

CLEAR &H7600

规定程序不能使用十六进制地址“7600”以前的内存空间，只能从 7600 开始向后存放。

注释：语句中的数值取值范围：十六进制数为 &H703E～&H77FF。

第七章
趣味程序集锦

本章讨论的内容分两部分：

第一部分简要介绍 F BASIC 的剩余指令：

背景、卡通符号库选择指令	CGEN
键控绘图画面调出指令	VIEW
发出蜂鸣声指令	BEEP
音乐旋律演奏指令	PLAY
测试光标水平位置指令	POS
测试光标垂直位置指令	CSRLIN
测试背景画面内容指令	SCR $
程序执行中断指令	STOP
程序指令中断解除指令	CONT
数据外存储指令	SAVE
数据外存储检查指令	LOAD?
数据从外存调入指令	LOAD
功能键定义指令	KEY
显示功能键定义指令	KEYLIST
向内存单元置数指令	POKE
从内存单元读数指令	PEEK
从键盘读入字符串指令	LINPUT
内存剩余单元测试指令	FRE
开关指令	ON GOTO

ON GOSUB
ON RETURN
ON RESTORE

返回 BASIC 菜单指令　　SYSTEM

调用机器语言子程序指令　　CALL

第二部分介绍十六个趣味程序，供大家编程参考和赏玩。

第一节　还有什么指令

本节将代替本章的指令小结，简要介绍 F BASIC 剩余的 22 条指令。

1. **背景、卡通符号库选择指令(CGEN)**

格式：CGEN<n>

类型：语句、命令

简写：CGE.

功用：选择背景、卡通显示画面使用的符号库。

注释：CGEN 为“Character GENeration”的缩写，是“产生字符或符号”的意思(“Character ”为字符的意思，“Generation”表示产生、发生的意思)。

n——符号库组合代码，取值范围为 0～3。它可以取常数、变量或表达式。

说明：

(1) 符号库的结构

F BASIC 有两个符号库，编号为“0、1”，每个库有 256 个 8×8 点阵的符号，编号为 0～255。0# 库为卡通符号库，主要存储了各种卡通的图形符号(见附录一)；1# 库则为背景符号库，主要存储了 104 个背景图形、ASCII 码字符和部分日文片

假名等(见附录二)。

(2) 符号库的使用

背景、卡通两个显示页都可以使用两个库的符号,因而有四种组合,分别由四个代码表示,其对应关系见表 7—1。

表 7—1 背景、卡通页选用符号库与代码对应一览表

符号库组合代码	背景使用库号	卡通使用库号	符号库使用说明
0	0	0	背景页使用卡通符号 卡通页使用卡通符号
1	0	1	背景页使用卡通符号 卡通页使用背景符号
2	1	0	背景页使用背景符号 卡通页使用卡通符号
3	1	1	背景页使用背景符号 卡通页使用背景符号

表 7—1 说明了 F BASIC 复位后系统自动设定为

```
CGEN  2
```

即背景、卡通两页各自使用自己图形库的状态。

当我们执行

```
CGEN  0
```

或

```
CGEN  1
```

后,将看到画面呈突起状,即使清屏也无济于事,这是什么原因呢?

从表 7—1 可知,符号库组合代码为"0、1"时,背景页使用卡通符号。由于背景页的清屏就是把整个画面填满空格(ASCII 码为 32),而卡通符号库中代码为 32 的图形正是丽

莎走图形头部的左半部，故产生了画面突起的效果。由此可见，使用 CGEN 语句令背景页使用卡通图形，可以打破同一水平坐标上显示卡通数不能超过 4 个的限制，而显示任意个卡通。因此，我们可以使用 256 个卡通图形任意组合出各种奇怪的、巨型的卡通，如一头四身八只脚的乌龟、身体中间爆炸的妖怪、玛丽骑企鹅、丽莎抓螃蟹等有趣的图案。大家尽可以发挥自己丰富的想象力，构造出你喜爱的卡通。

2. **键控绘图画面的调出指令(VIEW)**

格式：VIEW

类型：语句、命令

简写：V.

功用：调出 BG GRAPHIC 画面。

在程序的执行中，使用 VIEW 语句可以调出在 BG GRAPHIC 状态下绘制背景画面。

注释：VIEW 是“观看、观察”的意思。

VIEW 的实质是复制一份与 BG GRAPHIC 相同的画面。

演示：

(1)在 F BASIC 主菜单中选择第二项“BG GRAPHIC”进入键控绘图画面。

(2) 按键控绘图步骤绘出满意的图案。

(3) 绘图完毕，按“Esc”键后按“Pause”键，返回主菜单，选择第一项进入 BASIC 状态。

(4) 编写 BASIC 程序。

(5) 程序编写完毕，运行修改。一切正常后，在程序的“CLS”指令后加入“VIEW”语句。

(6) 运行程序，即可调出绘好的背景画面，并继续往下

执行程序。

示例：

```
10 VIEW
```

注——关于 BG GRAPHIC 键控绘图的操作方法将在丛书之七《电脑绘图》中详细介绍。

3. 发出蜂鸣声指令(**BEEP**)

格式：BEEP

类型：语句、命令

简写：B.

功用：控制扬声器发出 1000 赫兹的鸣叫声。

注释：BEEP 是尖锐短暂的音响的意思。

BEEP 语句可以在游戏程序中使用(比如作为达到或击中某一目标的回应)，也可以在程序的某处用来设计报警信号(如提示输入错误等)。

示例：

```
10 REM BEEP
20 INPUT "CONTINUE(Y/N)";A $
30 IF A $ ="Y" THEN 20
40 IF A $ ="N" THEN 100
50 BEEP
60 PRINT "PLEASE PRESS 'Y' OR 'N' KEY!"
70 GOTO 20
100 END
```

运行程序，若按“Y”键则要求继续输入；若按除“N、Y”外的其它键将发出鸣叫声，并显示“请按‘Y’或‘N’键”的提示，要求继续输入；若按“N”键则结束运行。

4. **音乐旋律演奏指令(PLAY)**

格式：PLAY“弦乐器资料”

类型:语句、命令

简写:PL.

功用:演奏乐曲。

注释:PLAY 为扮演、演奏的意思。

弦乐器资料为特定的字符串,由它规定了演奏中的音程、8 度音、音量、和音、主音、音色、节奏、休止等方式。使用冒号(:)可以演奏三重音。

示例:

```
PLAY "CRDRE"
```

可演奏‖ 1 0 2 0 3 ‖

弦乐资料的内容、结构比较复杂,将在丛书之七《电脑作曲》中详细讨论。

5. **测试光标的水平坐标指令(POS)**

格式:POS(N)

类型:函数

功用:测试光标当前的水平坐标(X)。

注释:POS 取自于"POSition",为位置的意思。

n——是一个哑变量,无实际意义,可取-32768～32767之间的任意值,一般取"0"。

示例:

```
10 REM POS
20 CLS
30 FOR I=0 TO 25: LOCATE I,0
40 PRINT POS(0): PAUSE 20
50 NEXT
```

运行程序,将从左向右依次打印"0、1、2、……25"。即光标的"X"坐标值。

6. **测试光标的垂直坐标指令(CSRLIN)**

格式:CSRLIN

类型:函数

简写:CSR.

功用:测试光标当前的垂直坐标(Y)。

注释:CSRLIN 取自于"CurSoR LINe",是光标所在行的意思。

示例:

```
10 REM CSRLIN
20 CLS
30 FOR I=0 TO 10
40 LOCATE I*2,I
50 PRINT POS(0)",",CSRLIN
60 PAUSE 20: NEXT
```

运行程序,将打印出

```
1, 0
  2, 1
    4, 2
      6, 3
        8, 4
          10, 5
            12, 6
              14, 7
                16, 8
                  18, 9
                    20, 10
```

7. **测试背景画面内容指令(SCR$)**

格式:SCR$(X,Y,S)

类型:函数

简写:SC.

功能:求出背景画面中所显示的文字或图画的数值。

当需要了解画面上某一位置(X,Y)处所显示的内容或该位置的配色号码时,使用 SCR $ 函数极为有效。

注释:SCR $ 源于"SCReen"(屏幕),"$ "表示字符串,意思是测试屏幕上的字符。

X——为测试点水平方向的坐标,取值范围为 0～27。它可以是常数、变量或表达式。

Y——为测试点垂直方向的坐标,取值范围为 0～23。它可以是常数、变量和表达式。

S——为求测试点的配色号码软开关。S=1 时表示开关打开,这时的 SCR $ 语句将测试出当前坐标处图形的配色号码;S=0 时表示关闭开关,不求配色号码,这时的 SCR $ 语句将测试出当前光标处图形的代码。当 S=0 时 S 参数可以省略。如:

SCR $ (X,Y)

示例:

```
10 REM SCR $
20 CLS
30 LOCATE 0,10: PRINT "FAMILY COMPUTER"
40 PRINT "....................."
50 LOCATE 10,15: PRINT SCR $ (0,10)
60 A $ =SCR $ (1,10): PRINT A $
70 C $ =SCR $ (1,10,1): PRINT "COLOR="ASC(C $ )
```

运行程序,将打印

FAMILY COMPUTER

.....

F

A

COLOR＝0

OK

■

8. **程序执行中断指令(STOP)**

格式:STOP

类型:语句

简写:STO.

功用:中断程序运行,将电脑置于 BASIC 的命令方式或立即方式。

中断运行的目的是为了调试程序,观察某些变量在程序运行期间的中间结果。

注释:STOP 是"停止、中断"的意思。

STOP 语句可以放在程序中的任何语句之后。程序运行中断后,电脑会以如下方式显示中断时 STOP 语句所在的行号:

BREAK IN ×××　　(×××表示中断的行号)

OK.　　("OK"后的小数点表示可以使用"CONT "指令从中断点处继续运行程序)

■

此时,用户可以检查变量的取值。然后可用"CONT"命令恢复程序运行。

(示例见"CONT"指令的介绍)

9. **程序指令中断解除指令**(CONT)

格式:CONT

类型:命令

简写:C.

功用:使中断的程序恢复运行。

注释:CONT 取自于“CONTinue”,为继续的意思。CONT 命令只适用于在程序运行中断后使用。运行中断后 BASIC 将给出断点位置(行号)。

示例:

```
10 REM STOP CONT
20 FOR I=1 TO 100
30 IF I=10 THEN STOP
40 PRINT I,
50 NEXT
```

运行程序,依次打印

```
1     2     3     4
5     6     7     8
9
BREAK IN 30
OK.
■
```

键入 CONT 命令,将继续打印

```
10    11    12    13
14    ..........
```

注意:程序运行中断后,若对程序进行了修改,则 CONT 命令无法恢复程序的继续运行。

10. **数据外存储指令(SAVE)**

格式 1:SAVE[“文件名”]

类型:命令

简写:SA.

功用:将指定的程序资料存入录音机的磁带中。

注释:SAVE 的英文意思是保存、存储。

存储程序时,要给程序起一个名字,称为文件名,以便于以后使用该程序。

注意:文件名必须用双引号括起来。

示例:

SAVE "TEST"	(存储 TEST 文件)
WRITING TEST	(提示正在存储 TEST 文件)
OK	(存储完毕)

格式 2:SAVE "A:文件名"

类型:命令

简写:SA.

功用:将指定的程序资料存入 3.5 英寸的磁盘中(只有在裕兴新世纪普及型电脑中才能使用)。

注意:保存文件时,必须先把磁盘插入软盘驱动器。文件名为存盘的文件名,由 1～8 个英文字母组成,不必输入扩展名,存盘后系统会自动以".BS"为扩展名加在文件名的后面,以于浮点 BASIC 或 LOGO 等文件相区别。

11. **数据外存储检查指令(LOAD?)**

格式:LOAD? ["文件名"]

类型:命令

简写:LO.? 或 LO.P.

功用:核对待保存的程序是否已正确地写入到磁带中。

程序写入磁带后,一般要进行是否正确写入的验证。若写

入的程序无误方可放心关机；若写入的程序有误，执行LOAD?命令后，会给出有错提示，这时需要把程序另写一遍，以保证程序的完整性。

注释：LOAD?源于“LOAD PRINT”，在此表示对磁带上的内容进行核对。

当指定文件名时，则在找到所确定的文件名程序之前，会跳过其它程序继续向下寻找。

当省略文件名时，则核对的是最初找到的文件。

示例：

```
LOAD?                  (无文件名的情况)
LOADING                (在核对中)
OK                     (核对正确)
■
LOAD?"TEST"            (有文件名的情况)
LOADING TEST           (核对到 TEST 文件)
OK                     (核对正确)
■
```

若显示

```
? TP ERROR             (核对结果有错)
```

需要重新写入一次。

12. **数据从外存调入指令(LOAD)**

格式 1：LOAD[“文件名”]

类型：命令

简写：LO.

功用：将程序从录音带中取出装入到系统内存中。

注释：LOAD 为载入的意思。

文件名为用 SAVE 命令保存程序资料时使用的名称。

说明：

(1) 用 LOAD 命令从录音机磁带调入程序资料前，应先将内存中的程序清除。

(2) 无论在使用 LOAD 还是 LOAD?，都应在按回车键后按录音机的放音键。

示例：

```
LOAD                    (不带文件名时调入第一个文件)
LOADING TEST
OK
■
LOAD"TEST－2"           (带文件名时)
SKIP TEST               (提示跳过 TEST 文件)
SKIP TEST－1            (提示跳过 TEST－1)
LOADING TEST－2         (提示正在调入 TEST－2)
OK                      (调入完毕)
■
```

注意：在使用录音机存储程序或图形时，由于录音机高频特性的差异，往往出现保存容易，调出困难的情况。因此，在保存和调入文件之前必须调整好录音机的音调、音量。最方便、最可靠的方法是使用录音机测试带进行调整，需要的朋友可参考丛书之四《中文 BASIC》中的录音机调试一节。

格式 2：LOAD "A：文件名"

类型：命令

简写：LO.

功用：将指定的程序资料从 3.5 英寸的磁盘中调入内存(只有在裕兴新世纪普及型电脑中才能使用)。

注意：调出文件时，必须先把磁盘插入软盘驱动器。文件

名为存盘的文件名，由 1～8 个英文字母组成，不必输入扩展名，调出时系统会自动寻找以“.BS”为扩展名的同名文件调入内存。

13. **功能键定义指令(KEY)**

格式：KEY n，“字符串”

类型：命令、语句

简写：K.

功用：对某一功能键进行重新定义。

当键盘重复输入或操作使用的单词或语句较多时，可使用 KEY 语句把单词或语句定义给某一功能键，以提高程序的输入速度和简化操作。

注释：KEY 在此为“键”的意思。

“n”的取值为 1～8。

“字符串”可以是除双引号之外的任意字符，一个功能键最多可定义 15 个字符。定义的字符必须用双引号括起来。

示例：

KEY 1，"COMPUTER"↙

把功能键“F1”定义为“COMPUTER”。

现在若按“F1”键，屏幕上将显示“COMPUTER”。

14. **显示功能键定义指令(KEYLIST)**

格式：KEYLIST

类型：命令、语句

简写：K. L.

功用：显示各功能键的定义内容。

在功能键的使用中，当某个功能键的定义不明确时，可使用 KEYLIST 命令列出各功能键的定义以便熟悉。

注释：KEYLIST 为“键列表”的意思。

示例：

KEYLIST↙

屏幕显示

F1 LOAD(M)
F2 PRINT
F3 GOTO
F4 CHR$(
F5 SPRITE
F6 CONT(M)
F7 LIST(M)
F8 RUN(M)

“(M)”表示按回车键。即当使用这些功能键输入命令时，不须按回车键就可执行。

KEY 指令使用的补充说明：

当定义带回车键的功能键时，定义语句的形式如下(如定义功能键“F1”为“KEYLIST(M)”)：

KEY 1,"KEYLIST"+CHR$(13)↙

“CHR$(13)”表示按回车键。

这时按“F1”键，将显示

F1 KEYLIST(M)
F2 PRINT
……

15. **向内存单元置数指令**(POKE)

格式：POKE<地址>,<数据>,<数据>,……

类型：命令、语句

简写：PO.

功用：把数据写入内存中某一地址开始的一个或连续几个单元。

注释:POKE 语句的具体功能是将数据放入地址所指定的内存单元中。地址可以取常数、变量或表达式,其取值范围为 -32768～32767(或 0～&HFFFF)。数据的取值范围为 0～255,它也可以取变量的形式。

说明:一个 POKE 语句可以向内存写入多个数据,每写入一个数据,内存地址会自动加一。

示例:

POKE 30720,11,12,13,14↙

执行后,30720～30723 单元的数据为

30720=11

30721=12

30722=13

30723=14

16. **从内存单元读数指令(PEEK)**

格式:PEEK(地址)

类型:语句

简写:PE.

功用:读出内存中某一单元的内容。

PEEK 语句中的地址是指内存中的读数地址,取值范围 -32768～32767,它可以是常数、变量或表达式。读出的数据范围为 0～255。

示例:

```
10 REM PEEK
20 FOR I=800 TO 820
30 PRINT "("I")="PEEK(I),
40 NEXT
```

运行程序,显示

(800)=254　　(801)=253　　(802)=32

(803)=255　　(804)=33　　(805)=106

……

17. 从键盘读入字符串指令(LINPUT)

格式:LINPUT"提示信息";<字符变量>

类型:语句、命令

简写:LIN.

功用:从键盘上接收整行字符的输入,并把它们当做一个字符串赋给串变量。

注释:LINPUT 是"Line INPUT"的缩写,是整行输入的意思。

LINPUT 指令与 INPUT 指令的相同之处是都从键盘上输入数据,不同的是 LINPUT 语句中的提示信息、标点符号都可当作资料赋给变量。另外 LINPUT 语句执行时不出现"?"号。

注意:使用 LINPUT 语句时,一次输入的字符数不要超过 31 个。否则将显示出错提示信息

? ST ERROR IN ××××

告诉你文字超过限额。

示例:

```
10 REM LINPUT
20 LINPUT"STRING=";A$
30 PRINT A$
40 GOTO 20
```

运行程序,显示

STRING=■　　输入　325,":↙

继续显示

STRING＝325,″：

说明将提示也作为信息赋给变量 A $。

18. **内存剩余单元测试指令(FRE)**

格式:FRE

类型:函数

简写:FR.

功用:报告用户内存中未被程序占用的空余字节数。

注释:FRE 取自“FREe”,是自由、空闲的意思。在此表示自由空间。

示例:

```
PRINT FRE↙
4030
```

表示内存的空余字节数为 4030 个。

19. **开关指令(ON)**

格式:ON＜表达式＞GOTO＜行号 1＞,＜行号 2＞,

……

ON＜表达式＞GOSUB＜行号 1＞,＜行号 2＞,

……

ON＜表达式＞RETURN＜行号 1＞,……

ON＜表达式＞RESTORE＜行号 1＞,……

类型:语句

简写:O. G.

O. GOS.

O. RE.

O. RES.

功用:根据表达式的取值大小,从多个分支中选择并执行一个分支。

注释：

ON—GOTO 是“当……时转去执行……行程序”的意思。

ON—GOSUB 是“当……时转去执行……行子程序”的意思。

ON—RETURN 是“当……时返回……行程序”的意思。

ON—RESTORE 是“当……时从……行开始恢复数据区指针”的意思。

表达式的值可以是数值变量，也可以是表达式。取值范围为 0～255，其值的大小决定着转移的方向。

行号是指各个执行目标的行号。

该语句根据表达式的取值决定执行目标。若其值为 1 则转移目标为“行号 1”，其值为 2 则转移目标为“行号 2”，以此类推。也就是说：表达式的取值与行号的位置一一对应。当表达式的值为 0 或大于 255 时，开关语句失效，程序顺序执行下一行语句。

示例：

```
10 REM ON－GOTO
20 INPUT"INPUT 1～4";A
30 ON A GOTO 100,200,300,400
40 PRINT"DATA ERROR!": GOTO 20
100 PRINT"DATA＝1": GOTO 20
200 PRINT"DATA＝2": GOTO 20
300 PRINT"DATA＝3": GOTO 20
400 PRINT"DATA＝4": GOTO 20
```

运行程序，显示

INPUT 1～4? ■

若键入“1”则打印“DATA＝1 ”；若键入“4”则打印“DATA＝

4 ”；

若键入的数据小于 1 或大于 4 则打印“DATAERROR!”。

```
10 REM ON-GOSUB ON-RETURN
20 INPUT"INPUT 1～3=";A
30 ON A GOSUB 100,200,300
40 PRINT"DATA ERROR": GOTO 20
100 PRINT "DATA=1": GOTO 500
200 PRINT "DATA=2": GOTO 500
300 PRINT "DATA=3"
500 ON A RETURN 700,800,900
600 END
700 PRINT"RETURN 1": END
800 PRINT"RETURN 2": END
900 PRINT"RETURN 3": END
```

运行程序，显示

INPUT 1～3? ■

若键入“2”则显示

DATA=2

RETURN 2

表示由 30 行控制调用 200 行子程序，并由 500 行控制返回 800 行。

```
10 REM ON-RESTORE
20 INPUT"INPUT 1～3=";A
30 ON A RESTORE 100,200,300
40 FOR I=1 TO 5: READ B: PRINT B,: NEXT
50 END
100 DATA 1,2,3,4,5
200 DATA 10,20,30,40,50
```

300 DATA 100,200,300,400,500

运行程序，显示

INPUT 1～3? ■

若键入“2”则显示

10　　20　　30　　40

50

20. **返回 BASIC 菜单指令(SYSTEM)**

格式:SYSTEM

类型:命令

简写:S.

功用:控制返回 F BASIC 菜单。

当需要进入 BG GRAPHIC 键控绘图状态绘图时或退出 F BASIC 系统时，键入 SYSTEM 可返回 F BASIC 菜单进行选项。

注释:SYSTEM 在英文中为“系统”的意思，在此表示系统菜单。

示例:

SYSTEM↙

将返回 F BASIC 菜单。

21. **调用机器语言子程序指令(CALL)**

格式:CALL<X>

类型:语句、命令

简写:CA.

功用:调用机器语言子程序，并向子程序传递参数。

机器语言具有运行效率高、占用内存少等优点，CALL 指令为 F BASIC 程序调用机器语言子程序提供了手段。

注释:CALL 为“叫、呼唤”的意思。

“X”为机器语言子程序的入口地址，取值范围为－32768～32767(或 0～&HFFFF)，也可以取变量或表达式的形式。

示例：

```
5 REM CALL
10 POKE &H6000,32,&H75,&HAB,&4C,15,&HB3
20 A=&H5FFF : B=1
30 CALL A+B
```

运行程序，将首先清屏，然后发出一声蜂鸣。这是调用了系统中的清屏和 BEEP 子程序的结果。

说明：关于机器语言子程序的编写和使用，我们将在丛书之六《汇编语言》中介绍。

至此，F BASIC BS. 2A 版本的操作指令全部介绍完毕。

第二节　格式规范的“九九表”

第二章我们曾详细讨论了“九九表”程序的编写方法，现在再给出格式规范的实现程序。程序清单见 No. 7－1。

```
10 REM No. 7－1, XIAO JIU JIU
20 CLS
30 FOR I=1 TO 9: FOR J=1 TO I
40 A=I*J: A$=STR$(A): A$=RIGHT$(A$,LEN(A
   $)－1)
50 IF LEN(A$)=1 THEN A$=A$+" "
60 PRINT RIGHT$(STR$(J),1);"X";RIGHT$(STR$(I),
   1);"=";A$;" ";
70 NEXT
80 PRINT: NEXT
```

运行程序，将打印出格式规范的“九九表”

```
1×1=1
1×2=2  2×2=4
1×3=3  2×3=6  3×3=9
1×4=4  2×4=8  3×4=12  4×4=16
1×5=5  2×5=10  3×5=15  4×5=20
5×5=25
1×6=6  2×6=12  3×6=18  4×6=24
5×6=30  6×6=36
1×7=7  2×7=14  3×7=21  4×7=28
5×7=35  6×7=42  7×7=49
1×8=8  2×8=16  3×8=24  4×8=32
5×8=40  6×8=48  7×8=56  8×8=64
1×9=9  2×9=18  3×9=27  4×9=36
5×9=45  6×9=54  7×9=63  8×9=72
9×9=81
OK
■
```

输出格式的规范，来源于使用了数据——字串转换函数“STR$”，去除了正数前面的空格。

第三节　勾股定理的验证

在直角三角形中，两直角边长的平方和等于斜边的平方，这就是著名的“勾股定理”。所有满足“勾股定理”的三个数据组成了勾股数组。勾股数组有无穷个，现给出斜边取100以内数值时，求所有勾股数组的程序，见No. 7－2。

```
5 REM No. 7－2, GOU GU SHU ZU
```

```
10 CLS: CLEAR: INPUT"A * A+B * B=C * C,INPUT C=";N
20 FOR A=3 TO N: IF A * 10>N * 7 THEN END
30 FOR B=A+1 TO N-1
40 FOR C=B+1 TO N
50 D=A * A: E=B * B: F=C * C
60 IF F<D+E TNEN 90
70 IF F>D+E THEN C=N: GOTO 90
80 PRINT A,B,C
90 NEXT: NEXT: NEXT
100 END
```

运行程序，显示

A * A+B * B=C * C,INPUT C=? ■

键入 100 后，回车，将打印出如下结果：

```
3     4     5
5     12    13
6     8     10
7     24    25
8     15    17
9     12    15
9     40    41
10    24    26
11    60    61
12    16    20
12    35    37
13    84    85
14    48    50
15    20    25
15    36    39
```

……

65　　72　　97

共 52 组勾股数。

第四节　算算看

一、水仙花数

数学上把一个 3 位数等于其每位数立方和的数称为水仙花数。如

153＝1×1×1＋5×5×5＋3×3×3

现给出求 100～1000 之间的所有水仙花数的程序，见 No. 7－3。

```
5 REM No. 7－3, SHUI XIAN HUA
10 CLS
20 FOR N＝100 TO 999
30 X＝N/100: Y＝(N－X * 100)/10: Z＝N－X * 100－Y * 10
40 IF N＝X * X * X＋Y * Y * Y＋Z * Z * Z THEN PRINT X,Y,Z
50 NEXT
60 END
```

运行程序，将打印出 4 个水仙数。

1　　5　　3

3　　7　　0

3　　7　　1

4　　0　　7

二、求素数

所谓“素数”，就是除了能被 1 和它本身整除外、不能被其

它任何整数整除的自然数。因此，判断一个自然数是否为素数的方法是：将某自然数“N”分别除以 2～N/2 之间的各个整数，若都不能整除，则“N”为素数。

根据这一思路，编写程序如 No. 7－4。

```
10 REM No. 7－4, SU SHU
20 CLS: CLEAR: INPUT"N=";N
30 FOR X=3 TO N+2 STEP 2: IF X>N THEN END
40 FOR I=2 TO X/2
50 IF X MOD I=0 THEN I=X/2: NEXT
60 NEXT
70 PRINT X;" ";
80 NEXT
90 END
```

运行程序，键入 100，回车后，将打印如下结果：

```
3  5  7  11  13  17  19  23
29 31 37 41  43  47  53  59
61 67 71 73  79  83  89  97
```

三、预测牛数

买来一头母牛，从第一年起，每年生一头小母牛，每头小母牛从第四个年头起，每年也生一头小母牛。问在第 20 年时，牛的总数是多少？

设 N1 表示当年出生的小母牛数，N2、N3 分别表示生长到第 2 年、第 3 年的小母牛数，N4 表示第 4 年和 4 年以上的母牛数。那么，对于某一年的 N4 应该是前一年的 N4 与 N3 之和，而 N3 则是前一年的 N2，N2 则是前一年的 N1，而 N1 又等于当年的 N4。所以，通过辗转赋值和套入循环可编写出程序 No. 7－5。

```
10 REM No. 7－5，YU CE NIU SHU
20 N4＝1：N1＝1：N2＝0：N3＝0
30 FOR I＝2 TO 20
40 N4＝N4＋N3：N3＝N2：N2＝N1：N1＝N4
50 IF I＜4 THEN 70
60 N＝N1＋N2＋N3＋N4：PRINT I,N
70 NEXT
80 END
```

运行程序，将打印出如下结果：

4	6
5	9
6	13
7	19
8	28
9	41
10	60
11	88
12	129
13	189
14	277
15	406
16	595
17	872
18	1278
19	1873
20	2745

若使用下标变量取代简单变量的辗转赋值，可对以上程序进行优化。优化后的程序见 No. 7－6。

```
10 REM No. 7－6
20 DIM A(20)：A(1)＝2：A(2)＝3：A(3)＝4
30 FOR I＝4 TO 20
40 A(I)＝A(I－1)＋A(I－3)：PRINT I,A(I)：NEXT
```

这可能是求牛数的最简程序了，你还能编写出更简的程序吗？

第五节　谁是罪犯

漆黑的夜晚，电闪雷鸣，风雨交加，三角花园五号住宅发生了一起凶杀案。经过公安人员细心排查，最后集中到5个可疑人身上。但尚未有这5个人与此案直接联系的证据，仅有如下几条线索可循（设5人的编号依次为X1、X2、X3、X4、X5）：

（1）若X1参与作案则X2也参与作案。

（2）X4、X5中至少一人参与作案。

（3）X2、X3中只有一人参与作案。

（4）X3和X4，要么同时作案，要么都与此案无关。

（5）如果X5参与作案，那么X1、X4也一定参与作案。

编写“谁是凶手”的判断程序。

1. **问题分析**　由于这一问题只有参与和不参与两种可能，所以，我们可以假设参与作案为“1”（真）；不参与作案为“0”（假）。因而可列出5个关系式：

根据线索（1）有关系式　X1＝X2

根据线索（2）有关系式　X4＋X5＞0

根据线索（3）有关系式　X2≠X3

根据线索（4）有关系式　X3＊X4＋（1－X3）＊（1－X4）＝1

根据线索(5)有关系式　X5＝X1＊X4

2. **编写程序**　由于5人都有作案的可能，所以引入五重循环编程如No. 7－7。

```
10 REM No. 7－7,SHUI SHI XIONG SHOU
20 FOR X1＝0 TO 1: FOR X2＝0 TO 1: FOR X3＝0 TO 1
30 FOR X4＝0 TO 1: FOR X5＝0 TO 1
40 IF X1<>X2 THEN 110
50 IF X4＋X5 <＝0 THEN 110
60 IF X2＝X3 THEN 110
70 IF X3＊X4＋(1－X3)＊(1－X4)<>1 THEN 110
80 IF X4＊X1<>X5 THEN 110
90 PRINT"X1 X2 X3 X4 X5"
100 PRINT X1;" ";X2;" ";X3;" ";X4;" ";X5
110 NEXT: NEXT: NEXT: NEXT: NEXT
120 END
```

运行程序，将打印如下结果

X1	X2	X3	X4	X5
0	0	1	1	0

结果表明，凶手是X3、X4两人。

第六节　大赛排名次

目前，国内举行的各种歌手、体育、知识等竞赛的评分办法，都是由若干个评分委员评分，然后去掉一个最高分，去掉一个最低分，求出其它几个评分的算术平均数。该平均数即为选手的得分。这种评分办法很合理，也很普及。使用F BASIC编程进行评分的程序如No. 7－8。

```
5 REM No. 7－8,PING FEN
```

```
10 CLS: INPUT"Xuan Shou shu=";M: DIM F(M),H(M)
20 INPUT"Ping Wei Shu=";N: A$="==========
   ==========="
30 FOR A=1 TO M: MAX=0: MIN=100: H(A)=A
40 PRINT"* * * * * * * * * * * * * * * * * * * * * * *
   *": PRINT A ;" hao xuan shou De Fen": PRINT
50 FOR B=1 TO N: PRINT B;" hao ping wei";: INPUT P
60 IF P>MAX THEN MAX=P
70 IF P<MIN THEN MIN=P
80 F(A)=F(A)+P: NEXT
90 F(A)=(F(A)-MAX-MIN)/(N-2): PRINT
100 PRINT "less a max" MAX: PRINT "less a min" MIN :
    PRINT A;" hao zui hou De Fen";F(A): PRINT A$
110 NEXT
120 FOR I=1 TO M-1: FOR J=I+1 TO M
130 IF F(J)>F(I) THEN SWAP F(I),F(J): SWAP H(I),H(J)
140 NEXT: NEXT
150 PRINT A$: PRINT
160 FOR I=1 TO M: PRINT "Di";I;" ming"; H( I) ; " hao";F
    (I);" fen"
170 NEXT: PRINT
180 PRINT A$: END
```

运行程序，显示

Xuan Shou shu=? ■

询问选手人数？假定有 5 名选手，键入 5，回车，显示

Ping Wei Shu=? ■

询问评委人数？假定有 7 名评委，键入 7，回车，显示

```
**********************
1 hao xuan shou De Fen   (1 号选手得分)
```

1 hao ping wei?　　（1 号评委打分）

假如 1 号评委打分 97 分，则键入 97。

以后将依次询问 2、3、……7 号评委的打分，假如依次打分 87、76、95、98、77、88，依次输入后，将显示

****

1 hao xuan shou De Fen

1 hao ping wei? 97　　（1 号评委打分 97）

2 hao ping wei? 87　　（2 号评委打分 87）

3 hao ping wei? 76　　……

4 hao ping wei? 95

5 hao ping wei? 98

6 hao ping wei? 77

7 hao ping wei? 88

less a max 98　　（丢掉一个最高分）

less a min 76　　（丢掉一个最低分）

1 hao zui hou De Fen 88　（1 号最后得分 88 分）

==========

随后，又做 2 号选手的评分准备。

若以后 4 名选手的得分依次为：

2 hao zui hou De Fen 85

3 hao zui hou De Fen 82

4 hao zui hou De Fen 83

5 hao zui hou De Fen 97

待评分完毕，立即打印比赛名次

==========

di 1 ming 5 hao 97 fen　（第一名 5 号选手 97 分）

di 2 ming 1 hao 88 fen　……

di 3 ming 2 hao 85 fen

di 4 ming 4 hao 83 fen

di 5 ming 3 hao 82 fen

==========

第七节　方便的电话号码册

若拥有一个带检索功能的电脑电话号码册，将给你带来诸多方便。程序见 No. 7－9。

```
5 REM No. 7－9, DIAN HUA HAO MA CE
10 CLS: INPUT "How many:"; Z: DIM A $ (Z, 2), B (Z) :
   PRINT
20 FOR I=1 TO Z: READ A $ (I,1), A $ (I,2): NEXT
30 FOR E=1 TO Z: S $ =LEFT $ (A $ (E,1),1): BE=ASC(S
   $ ) : NEXT
40 FOR F=1 TO Z－1
50 FOR G=1 TO Z－1
60 IF B(G)<B(G+1) THEN 80
70 SWAP B(G), B(G+1): SWAP A $ (G,1), A $ (G+1,1):
   SWAP A $ (G,2), A $ (G+1,2)
80 NEXT: NEXT
90 INPUT "please choose:LIST(1), LOOK(2), END(3) . INPUT
   1,2,3"; X
100 IF X=3 THEN END
110 IF X=1 THEN CLS: GOTO 170
120 CLS: INPUT"NAME"; T $ : PRINT
130 FOR R=1 TO Z
140 IF T $ =A $ (R,1) THEN PRINT T $, A $ (R,2): PRINT:
   GOTO 90
```

```
150 NEXT
160 PRINT"No this name!": PRINT: GOTO 90
170 PRINT"NAMES";: LOCATE 20,CSRLIN: PRINT "NUM-
    BERS": PRINT
180 FOR Q=1 TO Z: PRINT A$(Q,1);: LOCATE 20, CSR-
    LIN: PRINT A$(Q,2): PRINT: NEXT: PRINT: PRINT
190 GOTO 90
200 DATA Zhang xiao ming,327654,Li hua,781274, Wang ai
    hua,872357,Zhao zheng guo,871326,Qian shan qiang,626351
```

200行为姓名、电话号码数据语句，本程序只输入了5个人的电话号码，可一直向下输入多行数据语句。数据结构为“姓名，电话号码”，逐个向下排。

运行程序，电脑询问有多少个电话号码(How many:)，可输入数据语句中电话号码的总个数。现在输入5，回车后，显示

please choose:LIST(1),LOOK(2),END(3).

INPUT 1,2,3?

显示的内容是：“请选择：列出电话号码表(1)，查看某一个电话号码(2)，退出(3)。请输入1,2,3”。

若键入数字“1”，回车，则打印电话号码表如下：

NAMES	NUMBERS
Li hua	781274
Qian shan qiang	626351
Wang ai hua	872357
Zhang xiao ming	327654
Zhao zheng guo	871326,

若键入数字“2”，回车，则清屏，询问

NAME? ■ （输入查找的姓名）

如输入“Wang ai hua”，则打印

Wang ai hua 872357

若键入数字“3”，回车，则停机。

第八节　电脑猜谜

一、电脑能猜到你喜欢的日子

这是一个电脑猜数游戏，它能猜出你喜欢的日子。程序清单见 No. 7－10。

```
5 REM No. 7－10 CAI SHU
10 T=1: G=0
20 FOR A=1 TO 5: CLS: IF A>1 THEN T=T*2
30 LOCATE 9,2: PRINT"* * *";A;"* * *": V=6: H=0
40 FOR B=T TO 31 STEP T*2: FOR C=B TO B+T－1
50 H=H+5: IF H>20 THEN H=5: V=V+3
60 LOCATE H,V: PRINT C: NEXT: NEXT
70 LOCATE 0,20: PRINT"DO YOU LIKE? (1/0)";: F$=
   INKEY$(0): F=VAL(F$): G=G+T*F: NEXT
80 CLS: LOCATE 5,10: PRINT"YOUR FAVOR IS"G
90 LOCATE 3,20: PRINT"TRY AGAIN(Y/N)?";: A$=
   INKEY$(0): IF A$<>"N" THEN RUN
100 CLS: LOCATE 8,8: PRINT"GOODBYE!": LOCATE 0,
    22: END
```

运行程序，清屏，打印第一张表

```
    *   *   *   1   *   *   *
    1       3       5       7
```

9　　11　　13　　15
17　　19　　21　　23
25　　27　　29　　31

现在，你默记一个月中某一天的日期，观察第一张表中，是否有你喜欢的日期。若有你喜欢的日期则按数字键“1”；若无你喜欢的日期则按“0”键。

当按过“0”或“1”后，清屏，打印第二张表供选择。选择方法同第一张表。

当选择完五张表后，将打印出你喜欢的日期。

假如在第一张表中我们选择了“21”，则按“1”键，随后打印第二张表：

*　*　*　2　*　*　*
2　　3　　6　　7
10　　11　　14　　15
18　　19　　22　　23
26　　27　　30　　31

第二张表中无“21”，则按数字键“0”，接着打印第三张表：

*　*　*　3　*　*　*
4　　5　　6　　7
12　　13　　14　　15
20　　21　　22　　23
28　　29　　30　　31

第三张表中有“21”，按“1”键，打印第四张表：

*　*　*　4　*　*　*
8　　9　　10　　11

12	13	14	15
24	25	26	27
28	29	30	31

第四张表中无“21”，则按数字键“0”，接着打印第五张表：

*	*	*	5	*	*	*
16		17		18		19
20		21		22		23
24		25		26		27
28		29		30		31

第五张表中有“21”，按“1”键。清屏，显示

YOUR FAVOR IS 21

TRY AGAIN(Y/N)? ■

意思是：“你喜欢的日子是21号。再玩一次吗？”

若继续游戏则键入除“N”键外的任一键，将又从第一张表开始进行人机对话，玩下一轮游戏。若按“N”键，则清屏，显示

GOODBYE!

停机。

二、电脑查星期

我们知道1993年的国庆节是星期五，1994年的国庆节是星期六，也可以推出1995年的国庆节是星期日……。那么，2000年的国庆节是星期几呢？3000年呢？1949年开国大典日是星期几呢？芦沟桥事变日呢？八一南昌起义日呢？这些，推算出来要颇费一番脑筋。若运行程序No. 7—11，电脑可迅速回答你想了解的任一天是星期几。

```
5 REM No.7－11，CHA XING QI
10 CLS：DIM M(12),A$(6)
20 DATA 31,28,31,30,31,30,31,31,30,31,30,31
30 DATA SUNDAY(7),MONDAY(1),TUESDAY(2),
   WEDNESDAY(3),THURSDAY(4),FRIDAY(5),SATUR-
   DAY(6)
40 FOR I=1 TO 12：READ M(I)：NEXT
50 FOR J=0 TO 6：READ A$(J)：NEXT
60 INPUT "INPUT DATA：YEAR,MONTH,DAY=";Y,M,D
70 IF M=1 THEN 100
80 FOR I=1 TO M－1：C=C+M(I)：NEXT
90 IF (Y MOD 4=0) AND (Y MOD 100<>0) OR (Y MOD 100
   =0) AND (Y MOD 400=0) THEN C=C+1
100 YY=Y－1：S=YY+YY/4－YY/100+YY/400+C+D
110 S=S MOD 7
120 PRINT Y;".";M;".";D;".IS";A$(S)：PRINT
130 PRINT "PRESS ANY KEY TO BEGIN!"：B$=INKEY
    $(0) :IF B$="N" THEN 150
140 C=0：GOTO 60
150 END
```

运行程序，屏幕显示

```
INPUT DATA：YEAR,MONTH,DAY=? ■
```

要求输入查询星期的“年、月、日”(注意输入年份的数字时必须完整，如 1995 年不能简化为 95 年)，如输入

```
1992,10,1↙
```

打印

```
1992.10.1.IS THURSDAY(4)
PRESS ANY KEY BEGIN! ■
```

按除“N”键外的任一键将重新开始。

现在，我们若输入分别输入“1938，9，18”、“1949，10，1”、“2000，10，1”、“3000，10，1”。则分别打印

1938. 9. 18. IS WEDNESDAY(3)

1949. 10. 1. IS SATURDAY(6)

2000. 10. 1. IS SUNDAY(7)

3000. 10. 1. IS THURSDAY(4)

PRESS ANY KEY BEGIN! ■

三、电脑查血型

医学研究表明，子女的血型与父母的血型密切相关。法庭上也以父母、子女血型为依据判断子女属于亲生还是抱养。你知道自己的血型吗？你能根据你和你爱人的血型推知你的子女的血型吗？运行程序 No. 7－12，输入父母的血型就可以知道自己的可能血型了。当然输入你和你爱人的血型也可知道子女的可能血型。

```
5 REM No. 7－12, CHA XUE XING
10 CLS: DIM A$(4)
20 DATA A,B,AB,O
30 INPUT"father's=";F$: INPUT"mother's=";M$:PRINT"
   POSSIBLY=";
40 FOR I=1 TO 4: READ A$(I)
50 IF F$=A$(I) THEN U=I
60 IF M$=A$(I) THEN V=I
70 NEXT: IF U>V THEN SWAP U,V
80 ON U GOTO 90,110,120,140
90 IF V=1 OR V=4 THEN PRINT"A,O": END
100 IF V=2 THEN PRINT"A,B,AB,O": END
110 IF V=2 OR V=4 THEN PRINT"B,O": END
```

```
120 IF V=1 OR V=2 OR V=3 THEN PRINT"A,B,AB": END
130 IF V=4 THEN PRINT"A,B": END
140 IF V=4 THEN PRINT "O": END
150 PRINT"NONE!": END
```

运行程序，屏幕显示

father's=? ■（要求输入父亲的血型）

假设父亲的血型为“A”型，则键入“A”，回车，显示

mother's=? ■（要求输入母亲的血型）

假设母亲的血型为“B”型，则键入“B”，回车，显示

POSSIBLY=A,B,AB,O

表示子女的可能血型为“A”型、“B”型、“AB”型或“O”型。

若输入的父母血型分别为“A”、“O”型则显示

POSSIBLY=A,O

表示子女的血型可能为“A”型或“O”型。

程序 No.7－12 的编程依据为表 7－2，你能够根据表 7－2 编写出父母、子女血型互查的程序吗（即已知父母的血型查子女的可能血型、已知子女和父亲的血型反查母亲的可能血型、已知子女和母亲的血型反查父亲的可能血型）？

表 7－2　父母子女血型对照表

父　母　血　型		子女可能血型
O,A	A	O,A
B	A	O,AB,A,B
AB,A,B	AB	A,B,AB
O,B	B	O,B
O	AB	A,B
O	O	O

第九节　数字魔方

在中国古代“算经”中有一道“九宫”的数学题。题解说：

九宫之意，法似灵龟。

六八为肩，二四为足，

左七右三，戴九履一，

五居中央。

意思是说，用“1～9”九个自然数，组成一个三阶方阵，使方阵中每条边的数字之和、各对角线的数字之和均等于 15，数字排列见图 7－1。

2	9	4
7	5	3
6	1	8

图 7－1　九宫(一)

实质上，“九宫”就是一个边线等和的 3×3 方阵。能否将题意扩展，用电脑填出 5×5 或 7×7 的边线等和方阵呢？请见程序 No. 7－13。

```
5 REM No. 7－13,SHU ZI MO FANG
10 CLS: INPUT"INPUT AN ODD NUMBER";N: PRINT"N="
   ;N: PRINT
20 IF N MOD 2=0 THEN PRINT"ERROR:": PAUSE100: GO-
   TO 10
30 N1=N－1: DIM A(N1,N1)
40 I=0: J=N1/2: A(I,J)=1
50 FOR W=2 TO N * N
```

```
60 IF I<>0 THEN 80
70 L=N1: GOTO 90
80 L=I-1
90 IF J<>0 THEN 110
100 M=N1: GOTO 120
110 M=J-1
120 IF A(L,M)=0 THEN 140
130 I=I+1-(I+1)/N: GOTO 150
140 I=L: J=M
150 A(I,J)=W
160 NEXT
170 FOR I=0 TO N1: FOR J=0 TO N1: PRINT A(I,J);
180 P=P+A(I,J)
190 NEXT: PRINT: PRINT: NEXT
200 PRINT "THE SUM IS ";P/N
210 END
```

运行程序，键入“3”回车，将打印图 7－2 所示的方阵。

```
 6   1   8
 7   5   3
 2   9   4
THE SUM IS 15
```

图 7－2　九宫(二)

对比图 7－1，程序打印的方阵与九宫题解的结果是相同的(仅上下行颠倒了位置)。

若键入“5”则打印如图 7－3 所示的方阵。

```
 15   8   1  24  17
 16  14   7   5  23
 22  20  13   6   4
  3  21  19  12  10
  9   2  25  18  11
THE SUM IS 65
```

图 7—3　5 阶方阵

你可以计算一下，任意横、竖、斜线上的 5 个数字相加之和是否为 65。

下面列出 7 阶、9 阶方阵排列形式于图 7—4、7—5 供参考。

```
 28  19  10   1  48  39  30
 29  27  18   9   7  47  38
 37  35  26  17   8   6  46
 45  36  34  25  16  14   5
  4  44  42  33  24  15  13
 12   3  43  41  32  23  21
 20  11   2  49  40  31  22
THE SUM IS 175
```

图 7—4　7 阶方阵

```
45   34   23   12    1   80   69   58   47
46   44   33   22   11    9   79   68   57
56   54   43   32   21   10    8   78   67
66   55   53   42   31   20   18    7   77
76   65   63   52   41   30   19   17    6
 5   75   64   62   51   40   29   27   16
15   04   74   72   61   50   39   28   26
25   14    3   73   71   60   49   38   36
35   24   13    2   81   70   59   48   37

          THE SUM IS 369
```

图 7—5　9 阶方阵

若去掉 180 行的求和语句，该程序可求出 40 以内的任意阶奇数方阵。

求奇数阶方阵还有更简单的程序，你有办法化简吗？

你有能力编写出求偶数阶方阵的程序吗？

第十节　考秀才

一、组合一元钱

问题一：

用一元及一元以下面值的人民币组合一元钱，共有多少种组合方式。

乍看上去，这好象是一个简单的排列组合问题，不值得研究。但仔细分析发现，它远不是一般的排列组合问题。由于一元以下面值的人民币有 5 角、2 角、1 角、5 分、2 分、1 分，故使用的组合基数将分别是 1、2、5、10、20、50 六个数，而这六个数

之间又没有明显的直接联系，很难写出解题公式。若用人工逐个排列组合，费工费时还不可靠；若用数学方法求解也非易事。因此，最方便的方法就是让电脑按顺序把各个组合基数排一遍，处理大量的重复操作正是电脑的特长。

解决该问题的简单编程方法就是采用五重循环结构，把所有的基数全部试一遍，程序见 No. 7－14。按这种思路编程，虽然能求出结果，但程序的运行时间太长（约两个小时），效率太低。对循环结构的程序来说，欲提高程序的运行效率，一般要从减少循环层数、增大循环初值、缩小循环终值、加大循环步长四个方面着手，程序 No. 7－15 将给出最优化的程序，它的运行时间仅为 2.5 秒钟。为压缩篇幅，下面仅列出两个程序清单供使用，程序的具体优化过程请参见《F BASIC 语言与编程技巧》一书。

```
5 REM No. 7－14，YI YUAN QIAN (1)
10 CLS: PRINT"Please wait.....": K1=2
20 FOR I=0 TO 1: S=0
30 FOR J=0 TO 5: FOR K=0 TO 10
40 FOR L=0 TO 20: FOR M=0 TO 50
50 N=0
60 S=I*50+J*20+K*10+L*5+M*2+N
70 IF S<100 THEN 100
80 IF S=100 THEN K1=K1+1: IF K1 MOD 100=0 THEN
   PRINT "K1=";K1
90 GOTO 110
100 N=N+1: IF N<101 THEN 60
110 NEXT: NEXT: NEXT: NEXT: NEXT
120 PRINT "TOTAL=";K1
130
```

运行程序，将运算 126 分钟才打印出

TATOL＝4563

的结果。

优化后的程序清单如下：

```
5 REM No. 7－15, YI YUAN QIAN (2)
10 CLS: K1＝2: DIM A(3)
20 FOR I＝0 TO 1: A(1)＝I＊50
30 FOR J＝0 TO (100－A(1))/20: A(2)＝A(1)＋20＊J
40 FOR K＝0 TO (100－A(2))/10: A(3)＝A(2)＋10＊K
50 FOR L＝0 TO (100－A(3))/5
60 K1＝K1＋(100－A(3)－L＊5)/2＋1
70 NEXT: NEXT: NEXT: NEXT
80 PRINT "TOTAL＝";K1: END
```

运行程序，将立即打印出“TOTAL＝4563”的结果。

二、数红枣

问题二：

一篮红枣，若将它们分成三颗一组，余一颗；分成五颗一组，余两颗；分成七颗一组，余三颗。求这篮红枣共有多少颗。

程序见 No. 7－16。

```
5 REM No. 7－16, HONG ZAO
10 CLS: INPUT"Chushu＝";I: INPUT"Yushu＝";X
20 INPUT"Xia yi ge Chushu＝";J: IF J＜0 THEN 100
30 INPUT"Xia yi ge Yushu＝";Y
40 IF X MOD J＝Y THEN 60
50 X＝X＋I: GOTO 40
60 S＝I: T＝J
70 R＝S MOD T
```

```
80 IF R<>0 THEN S=T: T=R: GOTO 70
90 I=I*J/T: GOTO 20
100 PRINT "Xmin=";X
110 PRINT "X=K";CHR$(181);I;"+";X
120 END
```

程序 No. 7－16，必须在“下一个除数”询问时，输入小于“0”的数据才能结束输入。

运行程序，屏幕显示如下提示：

Chushu＝？ ■（除数是多少？）

键入“3”，回车后显示

Yushu＝？ ■（余数是多少？）

键入“1”，回车后显示

Xia yi ge Chushu＝？ ■（下一个除数是多少？）

键入“5”，回车后显示

Xia yi ge Yushu＝？ ■（下一个余数是多少？）

键入“2”，回车。

Xia yi ge Chushu＝？ ■（下一个除数是多少？）

键入“7”，回车后显示

Xia yi ge Yushu＝？ ■（下一个余数是多少？）

键入“3”，回车。

Xia yi ge Chushu＝？ ■（下一个除数是多少？）

键入“－1”，回车后将打印结果，显示

Xmin＝52

X＝K×105＋52

结果表示，一篮红枣的最小解是 52。其后凡是 105 的倍数加 52 的数字都是红枣的解（如 157、262 等）。

红枣问题在数学上称为余数问题，程序 No. 7－16 是余

数问题的通用求解程序。下面再使用该程序做两道余数问题。

三、算阶梯

问题三：

爱因斯坦曾做过这样一道数学题：一条阶梯，若每步跨两阶剩一阶；若每步跨三阶剩两阶；若每步跨四阶剩三阶；若每步跨五阶剩四阶；若每步跨六阶剩五阶；若每步跨七阶则不剩。问这条阶梯有多少阶？

运行程序，依次输入

2、1、3、2、4、3、5、4、6、5、－1，

回车后，将显示

Xmin＝119

X＝K×420＋119

四、求求未知数

问题四：

有物不知其数，九九数之剩二；八八数之剩三；七七数之剩四；六六数之剩五；五五数之剩四；四四数之剩三；三三数之剩二；二二数之剩一。问物若何？

运行程序，依次输入

9、2、8、3、7、4、6、5、5、4、4、3、3、2、2、1、－1

回车后，将打印

Xmin＝1019

X＝K×2520＋1019

第十一节　哥德巴赫猜想

著名数学家哥德巴赫(Goldbach,1690～1764)是德国人，1725年当选为俄国彼得堡科学院院士。哥德巴赫猜想是他在1742年与大数学家欧拉的通信中提出来的，其内容就是关于正整数和素数之间关系的两个推测。用通俗的语言描述就是：

1. **每一个不小于6的偶数都是两个奇素数之和。**

2. **每一个不小于9的奇数都是三个奇素数之和。**

这就是哥德巴赫猜想的内容，通常称为“偶数猜想”和“奇数猜想”。

在数论中，用符号{a,b}表示：任一个大偶数都可表示为一个不超过a个素因子的乘积和另一个不超过b个素因子的乘积之和。换言之，任何一个大偶数，可用a个素数的乘积与b个素数的乘积的和表示。如偶数20可表示为

$$\begin{aligned}20&=12+8\\&=2\times2\times3+2\times2\times2\end{aligned}$$

用数论符号表示就是{3,3}。若表示为

$$\begin{aligned}20&=18+2\\&=2\times3\times3+2\end{aligned}$$

就可表示为{3,1}。

可见，a、b的取值越小越接近猜想的要求。当a=1、b=1时，就完成了偶数猜想的证明。

下面列出哥德巴赫猜想证明的年代和进度：

1920后，有人证明了{9,9}；

1924年{7,7}被证明；

1932年{6,6}被证明；

1937 年{5,7}、{4,9}、{3,15}

被证明；

1938 年证明了{5,5}；

1939 年{4,4}被证明；

1956 年我国数学家王元证明了{3,4}；

1957 年又由王元证明了{2,3}；

1962 年由潘承洞证明了{1,5}；

1962 年王元、潘承洞分别证明了{1,4}；

1978 年由陈景润证明了{1,2}。

现在正一步步向{1,1}逼近。

下面介绍用 F BASIC 语言验证这两个猜想的程序，分别见程序 No. 7－17、程序 No. 7－18。

一、偶数猜想

偶数猜想的验证程序如下：

```
5 REM No. 7－17, Ou shu cai xiang
10 INPUT "Input an EVEN number";A: B=A/2
20 FOR Y=B TO A: X=A－Y
30 IF Y MOD 2=0 OR X MOD 2=0 THEN 180
40 IF X=3 THEN 140
50 IF X=1 THEN 180
60 W=Y: GOSUB 210: SY=P
70 W=X: GOSUB 210: SX=P
80 IF SX MOD 2=0 THEN 100
90 SS=SX+2: GOTO 110
100 SS=SX+1
110 N=3
120 IF Y MOD N=0 OR X MOD N=0 THEN 180
```

```
130 N=N+2: IF N<=SX THEN 120
140 N=SS
150 IF Y MOD N=0 THEN 180
160 N=N+2: IF N<=SY THEN 150
170 M=M+1: PRINT M,A;"=";X;"+"Y
180 NEXT
190 PRINT: PRINT "D(";A;")=";M
200 CLEAR: GOTO 10
210 P=0: K=2
220 IF K*K>W THEN P=K: RETURN
230 K=K+1: GOTO 220
240 END
```

运行程序，输入 20，则打印

```
1    20=7+13
2    20=3+17
D(20)=2
```

运行结果表明，偶数 20 可由两组素数相加组合，即它的偶数猜想有两个解。

若输入 100，则打印 6 个解如下：

```
1    100=47+53
2    100=41+59
3    100=29+71
4    100=17+83
5    100=11+89
6    100=3+97
D(100)=6
```

1994 年是一个偶数年，把 1994 输入后将打印出 27 对{1,1}的素数对。兹录于下，以便对照。

1　　1994＝997＋997

2　　1994＝907＋1087

3　　1994＝877＋1171

4　　1994＝823＋1171

5　　1994＝757＋1237

6　　1994＝691＋1303

7　　1994＝673＋1321

8　　1994＝613＋1381

9　　1994＝571＋1432

10　　1994＝547＋1447

11　　1994＝541＋1453

12　　1994＝523＋1471

13　　1994＝463＋1597

14　　1994＝397＋1597

15　　1994＝373＋1621

16　　1994＝367＋1627

17　　1994＝337＋1657

18　　1994＝331＋1663

19　　1994＝271＋1723

20　　1994＝241＋1753

21　　1994＝211＋1783

22　　1994＝193＋1801

23　　1994＝163＋1831

24　　1994＝127＋1867

25　　1994＝61＋1933

26　　1994＝43＋1951

27　　1994＝7＋1987

D(1994)=27

二、奇数猜想

奇数猜想的数学表达式为：对于一个不小于 9 的奇数有

N=P1+P2+P3

对于奇数 15 有

15=3+5+7

15=5+5+5

用数论符号表示为

T(15)=2

它表示 15 有两种组合方式。

奇数猜想的验证程序如下：

```
5 REM No. 5－18,Ji shu cai xiang
10 CLS: INPUT"Input ODD number";N
20 FOR X=3 TO N/3 STEP 2
30 I=2: W=X: GOSUB 90: II=P
40 IF X MOD I=0 THEN 70
50 I=I+1: IF I<=II THEN 40
60 A=0: B=B+1: GOSUB 120
70 NEXT
80 PRINT "T(";N;")=";M: END
90 P=0: K=2
100 IF K*K>W THEN P=K: RETURN
110 K=K+1: GOTO 100
120 R=N-X: Y=X
130 I=2: W=Y: GOSUB 90: II=P
140 IF Y MOD I=0 THEN 210
150 I=I+1: IF I<II THEN 140
```

```
160 Z=R-Y: I=2: W=Z: GOSUB 90: II=P
170 IF Z MOD I=0 THEN 210
180 I=I+1: IF I<=II THEN 170
190 A=A+1: M=M+1
200 PRINT M;B;A;N;"=";X;"+";Y;"+";Z
210 Y=Y+2: IF Y<=R/2 THEN 130
220 RETURN
```

运行程序，输入 33，将打印

```
1     1     1     33=3+7+23
2     1     2     33=3+11+19
3     1     3     33=3+13+17
4     2     1     33=5+5+23
5     2     2     33=5+11+17
6     3     1     33=7+7+19
7     3     2     33=7+13+13
8     4     1     33=11+11+11
T(33)=8
```

运行结果的第一列表示第几组解，T(33)=8；第二列表示 P1 的顺序号，“1”表示 P1 的第一个数“3”，“2”表示 P1 的第二个数“5”……；第三列表明一个确定的 P1 使用了几次，如 P1=3 使用了三次，即含“3”的有三组解，P1=5 使用了两次，说明含“5”的有两组解。

1995 年是个奇数年，运行程序，输入 1995，将打印出 2029 个解，其前 10 个解如下：

```
1     1     1     1995=3+5+1987
2     1     2     1995=3+13+1979
3     1     3     1995=3+19+1973
```

4　　1　　4　　1995＝3＋41＋1951

5　　1　　5　　1995＝3＋43＋1949

6　　1　　6　　1995＝3＋59＋1933

7　　1　　7　　1995＝3＋61＋1931

8　　1　　8　　1995＝3＋79＋1913

9　　1　　9　　1995＝3＋103＋1889

10　　1　　10　　1995＝3＋113＋1879

……

T(1995)＝2029

若输入 1997，将打印出 3309 个解

T(1997)＝3309

程序 No. 5－17、No. 5－18 证明了哥德巴赫的两个猜想确实存在。

第十二节　小企鹅进城

小企鹅笨的可爱，也呆的逗人。这不，它进城购物又出尽了洋相。本来提包随身带，买来的东西随手放进去最方便，但它却偏偏把包放到马路北边，再跑到路南买东西。更可笑的是它竟然带了 10 个提包。噢……！原来它还替 9 位朋友代买。10 个包分开放是怕把买的东西搞混了。你能说企鹅笨吗？但是，这下可够企鹅忙活的了。你看它跑到路南买一件，再跑到路北放一件，来往奔忙，马路差点被它踩成沟。呀！小心车辆！多危险啊！马路上的车太多了，小企鹅，你可要当心噢！

这是一个冒险类游戏，程序清单如下：

```
5 REM No. 5－19, QI E JIN CHENG
10 CLS: CLEAR: SPRITE ON: T＝300: T1＝9
```

```
20 LOCATE 2,0: PRINT "SCORE:";P
30 LOCATE 16,0: PRINT "LIFE:";T1
40 FOR I=0 TO 27: LOCATE I,1: PRINT CHR$(188): LO-
   CATE I,20: PRINT CHR$(188): NEXT
60 FOR I=1TO27 STEP3:LOCATE I,19:PRINT CHR$(209)
70 LOCATE I,3: PRINT CHR$(207):IF I=13 THEN I=11
80 NEXT
180 FOR I=0 TO 6: GOSUB 560: NEXT
190 XX=100: YY=180
200 GOSUB 570
210 I=0
220 X=XPOS(I): Y=YPOS(I)
230 IF X>220 THEN X=220: D=7: GOSUB 560
240 IF X<20 THEN X=20: D=3: GOSUB 560
250 IF ABS(X-XX)<8 AND ABS(Y-YY)<8 THEN PLAY"
    O3G1": T=1: GOTO400
260 I=I+1:IF I<7 THEN 220
270 S=STICK(0): IF S=0 THEN 320
280 E=-(S=1)*3-(S=2)*7-(S=8)-(S=4)*5: GO-
    SUB 570
320 XX=XPOS(7): YY=YPOS(7)
330 IF XX>226 THEN XX=226: GOSUB 570
340 IF XX<15 THEN XX=15: GOSUB 570
350 IF YY>167 THEN YY=167: GOSUB 570
360 IF YY<39 THEN YY=39: GOSUB 570
370 POSITION 7,XPOS(7),YPOS(7)
380 MOVE 7
390 X1=(XX-10)/8: Y1=(YY-4)/8-1
400 IF SCR$(X1,Y1)=CHR$(209) THEN GOSUB 600
410 IF SCR$(X1,Y1)=CHR$(207) THEN GOSUB 650
```

```
420 T=T-1
430 IF T<0 THEN T1=T1-1: T=300: LOCATE 21,0:
    PRINT T1
440 IF T1<1 THEN 490
445 IF P=1100 THEN 460
450 GOTO 210
460 PLAY"O3C2DGFEEDE": GOTO 500
490 PLAY"O3E1ECEC"
500 SPRITE OFF: CLS
510 FOR I=0 TO 7: ERA I: NEXT
520 LOCATE 8,8: PRINT "SCORE:";P
530 LOCATE 7,11: PRINT "PUSH START"
540 IF STRIG(0)=1 THEN 10
550 GOTO 540
560 DEF MOVE(I)=SPRITE(6,D,RND(3)+1,255,0,RND
    (3)): POSITION I,X,I*16+56: MOVE I: RETURN
570 DEF MOVE(7)=SPRITE(4,E,2,255,0,2): POSITION 7,
    XX,YY: RETURN
600 IF A<>1 THEN RETURN
610 PLAY"O4E0C"
620 LOCATE X1,Y1: PRINT CHR$(193)
630 P=P+100: LOCATE 8,0: PRINT P
640 A=0:RETURN
650 IF A<>0 THEN RETURN
660 LOCATE X1,Y1: PRINT " "
670 PLAY"O2F0"
680 P=P+10: LOCATE 8,0: PRINT P
690 A=1: RETURN
```

游戏方法与过程：

一人玩。用 0# 操纵器方向键控制企鹅运动(注意：企鹅的运动方向有惯性，即在改变企鹅的运动方向时，只要按一下方向键即可。若按住方向键不放，企鹅反而不能运动)。企鹅每次只能购买一个货物，购买货物后，只有返回马路对过，把货物放进提包后，才能进行第二次购物。10 个物品全部购齐并放进提包则为胜利。若货物没有购齐、或时间为零、或企鹅的命数为零则为失败。按 START 键后继续游戏。

附录一　卡通图形库代码一览表

卡通图形库代码一览表

10进制代码	16进制代码	图形说明	10进制代码	16进制代码	图形说明	10进制代码	16进制代码	图形说明
0	00		20	14		40	28	
1	01	玛丽走	21	15	玛丽	41	29	丽莎跳
2	02	1	22	16	背面	42	2A	
3	03		23	17		43	2B	
4	04		24	18		44	2C	
5	05	玛丽走	25	19		45	2D	
6	06	2	26	1A	玛丽跌	46	2A	丽莎滑
7	07		27	1B		47	2B	
8	08		28	1C		48	30	
9	09	玛丽走	29	1D	丽莎走	49	31	丽莎
10	0A	3	30	1E	1	50	32	背面
11	0B		31	1F		51	33	
12	0C		32	20		52	34	
13	0D	玛丽跳	33	21	丽莎走	53	35	丽莎跌
14	0E		34	22	2	54	36	
15	0F		35	23		55	37	
16	10		36	24		56	38	
17	11	玛丽滑	37	25	丽莎走	57	39	苍蝇
18	12		38	26	3	58	3A	1
19	13		39	27		59	3B	

卡通图形库代码一览表（续）

10进制代码	16进制代码	图形说明	10进制代码	16进制代码	图形说明	10进制代码	16进制代码	图形说明
60	3C		84	54		108	6C	
61	3D	苍蝇	85	55	飞鱼	109	6D	企鹅
62	3E	2	86	56	前2	110	6E	背面
63	3F		87	57		111	6F	
64	40		88	58		112	70	
65	41	飞鱼	89	59	妖怪	113	71	火球
66	42	1	90	5A	1	114	72	1
67	43		91	5B		115	73	
68	44		92	5C		116	74	
69	45	飞鱼	93	5D	妖怪	117	75	火球
70	46	2	94	5E	2	118	76	2
71	47		95	5F		119	77	
72	48		96	60		120	78	
73	49	飞鱼	97	61	企鹅走	121	79	车
74	4A	左上1	98	62	1	122	7A	左1
75	4B		99	63		123	7B	
76	4C		100	64		124	7C	
77	4D	飞鱼	101	65	企鹅走	125	7D	车左2
78	4E	左上2	102	66	2	126	7E	
79	4F		103	67		127	7F	
80	50		104	68		128	80	
81	51	飞鱼	105	69	企鹅	129	81	车
82	52	前1	106	6A	正面	130	82	左上1
83	53		107	6B		131	83	

卡通图形库代码一览表（续）

10进制代码	16进制代码	图形说明	10进制代码	16进制代码	图形说明	10进制代码	16进制代码	图形说明
132 133 134 135	84 85 86 87	车 左上2	156 157 158 159	9C 9D 9E 9F	杀手卫星左上	180 181 182 183	B4 B5 B6 B7	爆炸散发
136 137 138 139	88 89 8A 8B	车 上1	160 161 162 163	A0 A1 A2 A3	杀手卫星上	184 185 186 187	B8 B9 BA BB	乌龟走
140 141 142 143	8C 8D 8E 8F	车 上2	164 165 166 167	A4 A5 A6 A7	太空船左	188 189 190 191	BC BD BE BF	乌龟立正
144 145 146 147	90 91 92 93	太空站1	168 169 170 171	A8 A9 AA AB	太空船左上	192 193 194 195	C0 C1 C2 C3	螃蟹1
148 149 150 151	94 95 96 97	太空站2	172 173 174 175	AC AD AE AF	太空船上	196 197 198 199	C4 C5 C6 C7	螃蟹2
152 153 154 155	98 99 9A 9B	杀手卫星左	176 177 178 179	B0 B1 B2 B3	爆炸开始	200 201 202 203	C8 C9 CA CB	飞鸟1

卡通图形库代码一览表（续）

10进制代码	16进制代码	图形说明	10进制代码	16进制代码	图形说明	10进制代码	16进制代码	图形说明
204	CC	飞鸟2	222	DE	指示器1、2	242	F2	1～4
205	CD		223	DF		243	F3	
206	CE		224	E0	笔尖向上的羽毛笔	244	F4	
207	CF		225	E1		245	F5	阴体字COMPUTER
208	D0	激光弹1、2	226	E2		146	F6	
209	D1		227	E3		247	F7	
210	D2	激光束1、2	228	E4		248	F8	
211	D3		229	E5		249	F9	
212	D4	激光刀1、2	230	E6		250	FA	
213	D5		231	E7		251	FB	阴体字OPERATOR
214	D6	卡通用配色码1.2.3	232	E8	笔尖向下的羽毛笔	252	FC	
215	D7		233	E9		253	FD	
216	D8		234	EA		254	FE	
217	D9	音乐板使用的三音程	235	EB		255	FF	
218	DA		236	EC				
219	DB		237	ED				
220	DC	光标1、2	238	EE				
221	DD		239	EF				
			240	F0	黑体字			
			241	F1				

附录二　背景图形库代码一览表

背景图形库代码一览表

10进制代码	16进制代码	对应的符号	10进制代码	16进制代码	对应的符号	10进制代码	16进制代码	对应的符号
0	00	A1	20	14	C5	40	28	(
1	01	A2	21	15	C6	41	29	)
2	02	A3	22	16	C7	42	2A	*
3	03	A4	23	17	C8	43	2B	+
4	04	A5	24	18	D1	44	2C	,
5	05	A6	25	19	D2	45	2D	—
6	06	A7	26	1A	D3	46	2E	.
7	07	A8	27	1B	D4	47	2F	/
8	08	B1	28	1C	D5	48	30	0
9	09	B2	29	1D	D6	49	31	1
10	0A	B3	30	1E	D7	50	32	2
11	0B	B4	31	1F	D8	51	33	3
12	0C	B5	32	20	空格	52	34	4
13	0D	B6	33	21	!	53	35	5
14	0E	B7	34	22	"	54	36	6
15	0F	B8	35	23	#	55	37	7
16	10	C1	36	24	$	56	38	8
17	11	C2	37	25	%	57	39	9
18	12	C3	38	26	&	58	3A	:
19	13	C4	39	27	'	59	3B	;

背景图形库代码一览表（续）

10进制代码	16进制代码	对应的符号	10进制代码	16进制代码	对应的符号	10进制代码	16进制代码	对应的符号
60	3C	<	84	54	T	108	6C	l
61	3D	=	85	55	U	109	6D	m
62	3E	>	86	56	V	110	6E	n
63	3F	?	87	57	W	111	6F	o
64	40	@	88	58	X	112	70	p
65	41	A	89	59	Y	113	71	q
66	42	B	90	5A	Z	114	72	r
67	43	C	91	5B	[	115	73	s
68	44	D	92	5C	¥	116	74	t
69	45	E	93	5D	]	117	75	u
70	46	F	94	5E	^	118	76	v
71	47	G	95	5F	—	119	77	w
72	48	H	96	60	'	120	78	x
73	49	I	97	61	a	121	79	y
74	4A	J	98	62	b	122	7A	z
75	4B	K	99	63	c	123	7B	{
76	4C	L	100	64	d	124	7C	|
77	4D	M	101	65	e	125	7D	}
78	4E	N	102	66	f	126	7E	×
79	4F	O	103	67	g	127	7F	÷
80	50	P	104	68	h	128	80	ム
81	51	Q	105	69	i	129	81	メ
82	52	R	106	6A	j	130	82	モ
83	53	S	107	6B	k	131	83	ヤ

背景图形库代码一览表（续）

10进制代码	16进制代码	对应的符号	10进制代码	16进制代码	对应的符号	10进制代码	16进制代码	对应的符号
132	84	ユ	156	9C	ザ	180	B4	c
133	85	ヨ	157	9D	ジ	181	B5	×
134	86	ラ	158	9E	ズ	182	B6	÷
135	87	リ	159	9F	ゼ	183	B7	∽
136	88	ル	160	A0	ゾ	184	B8	E1
137	89	レ	161	A1	ダ	185	B9	E2
138	8A	ロ	162	A2	ヂ	186	BA	E3
139	8B	ワ	163	A3	ヅ	187	BB	E4 玛
140	8C	ン	164	A4	デ	188	BC	E5
141	8D	ヲ	165	A5	ド	189	BD	E6 丽
142	8E	マ	166	A6	バ	190	BE	E7
143	8F	イ	167	A7	ビ	191	BF	E8 兄
144	90	ウ	168	A8	ブ	192	C0	F1
145	91	エ	169	A9	ベ	193	C1	F2 弟
146	92	オ	170	AA	ボ	194	C2	F3
147	93	ヤ	171	AB	パ	195	C3	F4 用
148	94	ユ	172	AC	ピ	196	C4	F5
149	95	ヨ	173	AD	プ	197	C5	F6
150	96	ツ	174	AE	ペ	198	C6	F7
151	97	ガ	175	AF	ポ	199	C7	F8
152	98	ギ	176	B0	□	200	C8	G1
153	99	グ	177	B1	。	201	C9	G2 风
154	9A	ゲ	178	B2	[	202	CA	G3 景
155	9B	ゴ	179	B3	]	203	CB	G4

背景图形库代码一览表（续）

10进制代码	16进制代码	对应的符号	10进制代码	16进制代码	对应的符号	10进制代码	16进制代码	对应的符号
204	CC	G5	223	DF	I8	245	F5	L6 板
205	CD	G6 星	224	E0	J1 图	246	F6	L7
206	CE	G7	225	E1	J2	247	F7	L8 图
207	CF	G8 球	226	E2	J3 表	248	F8	M1
208	D0	H1	227	E3	J4	249	F9	M2 结
209	D1	H2 金	228	E4	J5 线	250	FA	M3 构
210	D2	H3 刚	229	E5	J6	251	FB	M4 用
211	D3	H4 二	230	E6	J7	252	FC	M5
212	D4	H5 代	231	E7	J8	253	FD	M6 方
213	D5	H6 用	232	E8	K1	254	FE	M7 块
214	D6	H7	233	E9	K2	255	FF	M8 用
215	D7	H8	234	EA	K3 单			
216	D8	I1	235	EB	K4			
217	D9	I2	236	EC	K5 图			
218	DA	I3 岛	237	ED	K6			
219	DB	I4	238	EE	K7 表			
220	DC	I5	239	EF	K8			
221	DD	I6	240	F0	L1 线			
222	DE	I7 双	241	F1	L2			
			242	F2	L3			
			243	F3	L4			
			244	F4	L5 平			

附录三　ASCII 码表

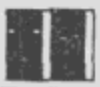
ASCII 码 一 览 表

10 进制代码	16 进制代码	对应的符号	10 进制代码	16 进制代码	对应的符号	10 进制代码	16 进制代码	对应的符号
32	20	空格	52	34	4	72	48	H
33	21	!	53	35	5	73	49	I
34	22	"	54	36	6	74	4A	J
35	23	#	55	37	7	75	4B	K
36	24	$	56	38	8	76	3C	L
37	25	%	57	39	9	77	4D	M
38	26	&	58	3A	:	78	4E	N
39	27	'	59	3B	;	79	4F	O
40	28	(	60	3C	<	80	50	P
41	29	)	61	3D	=	81	51	Q
42	2A	*	62	3E	>	82	52	R
43	2B	+	63	3F	?	83	53	S
44	2C	,	64	40	@	84	54	T
45	2D	—	65	41	A	85	55	U
46	2E	.	66	42	B	86	56	V
47	2F	/	67	43	C	87	57	W
48	30	0	68	44	D	88	58	X
49	31	1	69	45	E	89	59	Y
50	32	2	70	46	F	90	5A	Z
51	33	3	71	47	G	91	5B	[

ASCII 码 一 览 表 (续)

10 进制代码	16 进制代码	对应的符号	10 进制代码	16 进制代码	对应的符号	10 进制代码	16 进制代码	对应的符号
92	5C	¥	104	68	h	116	74	X
93	5D	]	105	69	i	117	75	Y
94	5E	^	106	6A	j	118	76	Z
95	5F	—	107	6B	k	119	77	{
96	60	'	108	6C	l	120	78	x
97	61	a	109	6D	m	121	79	y
98	62	b	110	6E	n	122	7A	z
99	63	c	111	6F	o	123	7B	{
100	64	d	112	70	p	124	7C	V
101	65	e	113	71	q	125	7D	}
102	66	f	114	72	r	126	7E	×
103	67	g	115	73	s	127	7F	÷

附录四　背景图形与代码对照表

	0	1	2	3	4	5	6	7
A								
B								
C								
D								
E								
F								
G								
H								
I								
J								
K								
L								
M								

附录五　F BASIC(BS. 2A 版本)出错信息一览表

错误代码	出错表示	错误信息	出错说明
0	NF	NEXT without FOR	无 FOR 却有 NEXT
1	SN	Syntax error	语法错误
2	RG	RETURN without GOSUB	无 GOSUB 却有 RETURN
3	OD	Out of DATA	READ 读的资料未备足于 DATA 中
4	IL	ILlegal function call	函数的调用不合法
5	OV	OVerflow	演算结果超过许可的范围
6	OM	Out of Memory	内存不足
7	UL	Undefined Line Number	无 GOTO、GOSUB、IF 所指的行号
8	SO	Subscript Out or range	数组下标超出规定
9	DD	Duplicate Definition	数组双重定义
10	DZ	Division by Zero	用 0 作除数
11	TM	Type Mismatch	变量的类型不一致
12	ST	String Too long	字串超过 31 个字符
13	FT	Formula Too complex	式子过于复杂,如()太多
14	CC	Can't Continue	CONT 无法继续执行程序
15	UF	Undefined Function	使用了未定义的函数
16	MO	Missing Operand	指令未指定所需要的参数
17	TP	Tape read error	无法从磁带中读取资料

附录六　F BASIC(BS. 2A 版本)指令汇总表

一、命令一览表

命　　令	功　　能
CLEAR &HXXXX	规定 BASIC 程序的使用范围
NEW	清除内存中的程序
LIST [[M],[—N]]	将内存中的程序按行号大小列表
RUN [N]	执行内存中的程序
CONT	使用 STOP 指令中断的程序继续运行
LOAD ["文件名"]	从磁带中读出程序
SAVE ["文件名"]	把程序写入磁带中
LOAD? ["文件名"]	核对保存在磁带中的程序是否正确
CALL XXXX	调用机器语言子程序
PRINT [],[], PRINT [];[]; PRINT [] PRINT &HXXXX	分栏打印计算结果或数据资料 紧凑打印计算结果或数据资料 换行打印计算结果或数据资料 把 16 进制数据转化为 10 进制数输出
KET N,"字符串"	设定功能键
KEYLIST	显示各功能键的定义

二、一般语句一览表

语　　句	功　　能
PRINT [],[],	分栏打印计算结果或数据资料
PRINT [];[];	紧凑打印计算结果或数据资料
PRINT []	换行打印计算结果或数据资料
INPUT ["提示信息"];变量[,……]	显示问号，从键盘上读入数据
INPUT ["提示信息"],变量[,……]	不显示问号，从键盘上读入数据
LINPUT ["提示信息"];字符变量	读取一整行数据，包括段落分隔符
LINPUT ["提示信息"],字符变量	号在内
CLEAR	清除变量的取值、解除数组定义
DIM 变量名(M1[,M2])[,……]	定义数组
GOTO N	无条件转移到由 N 指定的行号
GOSUB N	调用由 N 指定行号的子程序
RETURN	由子程序返回调用程序
RETURN N	由子程序返回到指定的行号 N
IF 表达式 THEN 行号	
IF 表达式 GOTO 行号	条件转移，用来判断并执行某种
IF 表达式 THEN GOTO 行号	功能
IF 表达式 THEN 语句	
FOR ～ TO ～ STEP	循环语句
NEXT	
ON 表达式 GOTO N1,N2,……	开关语句，无条件转向到系列行号
ON 表达式 GOSUB N1,N2,……	开关语句，调用系列行号的子程序
ON 表达式 RETURE N1,N2,……	开关语句，子程序返回到系列行号
ON 表达式 RESTORE N1,N2,……	开关语句，恢复数据区到系列行号
STOP	中断正在执行中的程序

语　　句	功　　能
END	结束程序运行
SWAP 变量1,变量2	交换两变量中的内容
REM [叙述]	对程序进行注释或说明
READ 变量1[,变量2……] DATA 数据1[,数据2……] RESTORE [行号]	从 DATA 语句中读取数据 为 READ 语句提供数据 恢复数据区指针到指定的行号
POKE 地址,数据1[,数据2……]	将数据存入指定的内存单元中
BEEP	使扬声器发出鸣叫声
PLAY "弦乐资料"	演奏音乐旋律
PAUSE N PAUSE	程序运行中暂停 N 时间 程序运行中暂停,按任意键后继续

三、一般函数指令

函　　数	功　　能
ABS (X)	对 X 取绝对值
SGN (X)	判别 X 的符号
RND (X)	产生 X 以内的随机整数
ASC (字符)	将字符转换成 ASCII 码
CHR $ (X)	求 X 对应的字符或图形
VAL (字符)	将字符串转换成10进制数据
STR $ (X)	将10进制数据转换成字符串
HEX $ (X)	将10进制数据转换成16进制数据
LEFT $ (字符串,N)	从字符串左边取 N 个字符形成子串
RIGHT $ (字符串,N)	从字符串右边取 N 个字符形成子串
MID $ (字符串,N,M)	从字符串的 N 位取 M 个字符形成子串
LEN (字符串)	测试字符串的长度
PEEK (X)	取指定内存地址 X 中的数值

四、特殊函数指令一览表

函　　数	功　　能
POS (0)	测试光标当前的水平位置坐标
CSRLIN	测试光标当前的垂直位置坐标
FRE	测试内存单元的空余数
STICK (X)	测试操纵器的方向键值
STRIG (X)	测试操纵器的功能键值
SCR$ (X,Y)	测试背景面(X,Y)处的图形代码
SCR$ (X,Y,S)	测试背景面(X,Y)处的图形配色
INKEY$ [0]	读取按键值

五、画面、背景控制指令一览表

指　　令	功　　能
LOCATE X,Y	指定显示位置的坐标
COLOR X,Y,S	在屏幕的(X,Y)位置染S号颜色
CGEN (N)	选定背景页、背景页的符号库
CGSET [M][,N]	指定背景、背景页的颜色组合
CLS	清屏
PALETB N,C1,C2,C3,C4	定义背景页的配色代码和底背景色
PALETS N,C1,C2,C3,C4	定义卡通页的配色代码和底背景色
SYSTEM	从BASIC状态返回选择菜单
VIEW	调出BG GRAPHIC绘制的画面
DEFMOVE (N) = SPRITE (A,B,C,D,E,F)	定义运动背景
MOVE N0[,N1,……]	启动运动背景运动
CUT N0[,N1,……]	令运动中的运动背景停止
EAR N0[,N1,……]	令运动中的运动背景停止并消失
POSITION N,X,Y	规定N号运动卡通的运动起点坐标
XPOS (N)	测试N号运动背景的水平坐标

61507

指　　令	功　　能
YPOS (N)	测试 N 号运动背景的垂直坐标
MOVE (N)	判断N 号运动背景的运动是否结束
DEF SPRITEN ,(A ,B ,C ,D ,E) =字符表达式	定义 N 号显示背景
SPRITE N,X,Y	定义 N 号显示背景的显示坐标
SPRITE N	令 N 号显示背景消失
SPRITE ON	打开背景显示页
SPRITE OFF	关闭背景显示页

附录七　背景图形库(封二)
附录八　系统配色代码表(封三)
附录九　卡通图形库(封四)